PanaXSeries

The One to Watch for Constant Innovation-Making the Future Come Alive

MICROCOMPUTER MN102H

MN102H74G/74F/74D/F74G
LSI User's Manual

Pub.No.22374-014E

Panasonic

PanaX Series is a trademark of Matsushita Electric Industrial Co., Ltd.
The other corporation names, logotype and product names written in this book are trademarks or registered trademarks of their corresponding corporations.

Request for your special attention and precautions in using the technical informaition and semiconductors described in this book

(1) An export permit needs to be obtained from the competent authorities of the Japanese Government if any of the products or technologies described in this book and controlled under the "Foreign Exchange and Foreign Trade Law" is to be exported or taken out of Japan.

(2) The contents of this book are subject to change without notice in matters of improved function.When finalizing your design, therefore, ask for the most up-to-date version in advance in order to check for any changes.

(3) We are not liable for any damage arising out of the use of the contents of this book, or for any infringement of patents or any other rights owned by a third party.

(4) No part of this book may be reprinted or reproduced by any means without written permission from our company.

(5) This book deals with standard specification. Ask for the latest individual Product Standards or Specifications in advance for more detailsd infomation required for your design, purchasing and applications.

If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book.

Chapter 1 General Description 1

Chapter 2 CPU Basics 2

Chapter 3 Bus Interface 3

Chapter 4 Interrupts 4

Chapter 5 Timers 5

Chapter 6 Serial Interface 6

Chapter 7 Analog Interface 7

Chapter 8 USB 8

Chapter 9 DMA 9

Chapter 10 System Control 10

Chapter 11 Ports 11

Chapter 12 Appendix 12

Contents

Chapter1 General Description

1-1 General Description ..... I - 2
1-1-1 Introduction ..... I - 2
1-1-2 Features ..... I - 2
1-1-3 Overview ..... I - 5
1-2 Basic Specification ..... I - 9
1-3 Block Diagtram ..... I - 10
1-4 Pin Function ..... I - 12
1-4-1 Single-chip Mode ..... I - 12
1-4-2 Memory Expansion Mode ..... I - 13
1-4-3 Processor Mode ..... I - 14
1-4-4 List of Pin Functions ..... I - 15
1-5 Adressing Mode ..... I - 29
1-6 List of Instructions ..... I - 30

Chapter2 CPU Basics

2-1 Machine Clock ..... II - 2
2-1-1 Machine Clock ..... II - 2
2-2 Instruction Excution Controller ..... II - 4
2-2-1 Configuration ..... II - 4
2-2-2 Pipeline Process ..... II - 5
2-3 Internal Registers ..... II - 7
2-3-1 Registers for Adrress ..... II - 7
2-3-2 Registers for Operation ..... II - 8
2-3-3 Processor Status Word ..... II - 10
2-4 Special Functions Registers ..... II - 13
2-5 Standby Function ..... II - 14
2-5-1 Overview ..... II - 14
2-5-2 CPU Mode Control Register ..... II - 16
2-5-3 Transfer between NORMAL Mode and SLOW Mode ..... II - 17
2-5-4 Transfer to Standby Mode ..... II - 19
2-6 Error Detection Function ..... II - 22
2-7 Reset Function ..... II - 23
2-8 CPU Special Register ..... II - 25

Chapter3 Bus Interface

3-1 Memory ..... III - 2
3-1-1 Internal Momery Location Examples ..... III - 2
3-2 Summery of Bus Interface ..... III - 4
3-2-1 Configuration ..... III - 4
3-2-2 Control Registers ..... III - 5
3-2-3 Transfer Bus Width Selection ..... III - 12
3-3 External Bus Connection Examples ..... III - 13
3-3-1 External Bus Connection Examples ..... III - 13
3-4 DMA Support Function ..... III - 24
3-4-1 Bus Arbitration ..... III - 24

Chapter4 Interrupts

4-1 Interrupt Controller ..... IV - 2
4-1-1 Overview ..... IV - 2
4-1-2 Control Registers ..... IV - 7
4-1-3 Interrupt Level ..... IV - 12
4-1-4 External Interrupt ..... IV - 15
4-1-5 NNMI Pin Interrupt ..... IV - 17
4-1-6 Interrupt Acceptance ..... IV - 18
4-1-7 External Interrupt Setup Example ..... IV - 20
4-1-8 Watchdog Timer Interrupt Setup Example ..... IV - 23
4-1-9 Extended Watchdog Timer Setup Example ..... IV - 26
4-1-10 Interrupt Return ..... IV - 28
4-1-11 Multiple Interrupt ..... IV - 29
4-2 Interrupt Control Register ..... IV - 31

Chapter5 Timers

5-1 Summery of 8-bit Timer ..... V - 2
5-1-1 List of 8-bit Timer Function ..... V - 2
5-1-2 Summery of Timer 0 and Timer 1 ..... V - 3
5-1-3 Timer 0 and Timer 1 Block Diagram ..... V - 5
5-1-4 Prescaler 0 Block Diagram ..... V - 6
5-1-5 Timer 2 to Time 9 Block Diagram ..... V - 7
5-2 8-bit Timer Control Registers ..... V - 10
5-2-1 List of Timer 0, Timer 1 and Prescaler 0 Control Registers ..... V - 10
5-2-2 Timer 0, Timer 1 and Prescaler 0 Programmable Timer Registers ..... V - 11
5-2-3 Timer 0, Timer 1 and Prescaler 0 Mode Registers ..... V - 13
5-2-4 List of Timer 2 to Timer 9 Control Rgisters ..... V - 15
5-2-5 Timer 2 to Timer 9 Programmable Timer Registers ..... V - 16
5-2-6 Timer 2 to Timer 9 Mode Registers ..... V - 20
5-3 Timer 0 and Timer 1 Setup Example ..... V - 29
5-3-1 Event Counter Using Timer 0 or Timer 1 ..... V - 29
5-3-2 Clock Output Using Timer 0 or Timer 1 ..... V - 31
5-3-3 Interval Timer Using Timer 0 or Timer 1 ..... V - 34
5-4 Timer 2 to Timer 9 Setup Example ..... V - 38
5-4-1 Interval Timer and Timer Output ..... V - 38
5-4-2 Interval Timer and Timer Output Setup Example ..... V - 41
5-5 Event Count (Timer 2 to Timer 9) ..... V - 43
5-5-1 Event Count ..... V - 43
5-5-2 Event Counter Setup Example ..... V - 45
5-6 Cascade Connection (Timer 2 to Timer 9) ..... V - 47
5-6-1 Cascade Operation ..... V - 47
5-6-2 Cascade Setup Example ..... V - 52
5-7 Summery of 16-bit Timer ..... V - 54
5-7-1 List of 16-bit Timer Function ..... V - 54
5-7-2 16-bit Timer Block Diagram ..... V - 55
5-8 16-bit Timer Control Register ..... V - 57
5-8-1 List of Timer 10 to Timer 13 Control Registers ..... V - 57
5-8-2 Timer 10 to Timer 13 Programmable Timer Registers ..... V - 58
5-8-3 Timer 10 to Timer 13 Mode Registers ..... V - 63
5-9 16-bit Timer Operation ..... V - 69
5-9-1 16-bit Timer Operation Examples ..... V - 69
5-9-2 Interval Timer Operation Examples ..... V - 85

Chapter6 Serial Interface

6-1 Serial Interface ..... VI - 2
6-1-1 Serial Interface Function ..... VI - 2
6-1-2 Serial Interface Block Diagram ..... VI - 3
6-2 Control Registers ..... VI - 4
6-2-1 List of Serial Interface Control Registers ..... VI - 4
6-2-2 Serial Reception Registers/Serial Transmission Registers ..... VI - 5
6-2-3 Serial Control Registers ..... VI - 6
6-2-4 Serial Interface Status Registers ..... VI - 7
6-2-5 Serial Status 1 Register ..... VI - 8
6-3 Serial Interface Operation ..... VI - 9
6-3-1 Serial Interface Operation ..... VI - 9
6-3-2 Serial Interface 0 Setup Example ..... VI - 13
6-3-3 I^2C Transmission (Serial 0 and Serial 1) ..... VI - 19
6-3-4 I^2C Reception (Serial 0 and Serial 1) ..... VI - 21

Chapter7 Analog Interface

7-1 Analog Interface ..... VII - 2
7-1-1 Analog Interface Function ..... VII - 2
7-1-2 Analog Interface Configuration ..... VII - 3
7-2 Control Registers ..... VII - 4
7-2-1 List of Analog Inteface Control Registers ..... VII - 4
7-2-2 A/D Conversion Control Register ..... VII - 5
7-2-3 A/D Conversion Data Buffer ..... VII - 6
7-3 A/D Converter Operation ..... VII - 9
7-3-1 A/D Converter Operation ..... VII - 9
7-3-2 A/D Operation Setup Example ..... VII - 14

Chapter8 USB

8-1 Description ..... VIII - 2
8-1-1 USB Function ..... VIII - 2
8-2 Features ..... VIII - 3
8-3 Block Diagram ..... VIII - 4
8-4 USB Special Functions ..... VIII - 5
8-4-1 Double Buffering Function ..... VIII - 5
8-4-2 Data Rate Feedback Function ..... VIII - 6
8-5 ATC Interface ..... VIII - 8
8-5-1 ATC Transfer Mode ..... VIII - 10
8-5-2 Packet Transfer Using Arbitration Timing Control ..... VIII - 12
8-6 Programming Considerations ..... VIII - 14
8-6-1 Interrupt Triggers ..... VIII - 14
8-6-2 Interrupt Processing ..... VIII - 16
8-6-3 Processing Multiple Interrupts within the Same USB Frame ..... VIII - 18
8-6-4 Acknowledging Interrupts from the Core ..... VIII - 18
8-6-5 Servicing Multiple Endpoints within the Same Interrupt ..... VIII - 19
8-6-6 Controlling the USB Core for Control Transfers ..... VIII - 20
8-6-7 Handling Suspend and Resume ..... VIII - 24
8-7 Source Code Examples ..... VIII - 25
8-7-1 Short Control Transfer ..... VIII - 25
8-7-2 OUT Toggle Test on Endpoint 4 ..... VIII - 26
8-7-3 Isochronous OUT Error Test on Endpoint 3 ..... VIII - 27
8-7-4 Bulk OUT Error Test on Endpoint 3 ..... VIII - 28
8-8 Internal Registers ..... VIII - 29
8-8-1 Function Adress Register ..... VIII - 31
8-8-2 Power Management Register ..... VIII - 32
8-8-3 Interrupt Registers ..... VIII - 33
8-8-4 Interrupt Enable Registers ..... VIII - 36
8-8-5 Index Registers ..... VIII - 38
8-8-6 Frame Number Registers ..... VIII - 39
8-8-7 IN Control Status Registers ..... VIII - 40
8-8-8 OUT Control Status Registers ..... VIII - 45
8-8-9 Endpoint 0 Control Status Register ..... VIII - 48
8-8-10 Maximum Packet Siza Register ..... VIII - 51
8-8-11 OUT Write Count Registers ..... VIII - 52
8-8-12 Sampling Buffes ..... VIII - 53
8-8-13 ATC Switch Register ..... VIII - 54

Chapter9 DMA

9-1 Sunnery of ATC Function ..... IX - 2
9-1-1 ATC Function ..... IX - 2
9-1-2 ATC Transfer Block Diagram ..... IX - 3
9-2 ATC Control Registers ..... IX - 4
9-3 ATC Operation ..... IX - 16
9-3-1 Software Activation/Hardware Activation ..... IX - 16
9-3-2 Transfer Sequence ..... IX - 17
9-3-3 Transfer Sequence for Multiple Channels ..... IX - 19
9-3-4 Single Adress Mode/Dual Adress Mode ..... IX - 20
9-3-5 C-Bus Mode, Early Read Mode and Early Write Mode ..... IX - 22
9-3-6 1 Cycle Operation Mode ..... IX - 24
9-3-7 Wait Control ..... IX - 25
9-3-8 ATC Transfer ..... IX - 27
9-3-9 ATC Transfer in Single Adress Mode ..... IX - 33
9-3-10 External Bus Request Arbitration ..... IX - 34
9-3-11 Transfer Suspend Using Nonmaskable Interrupt (NMI) ..... IX - 35
9-4 ATC Transfer Setup Example ..... IX - 36
9-4-1 DMA Transfer by Software Activation ..... IX - 36
9-4-2 DMA Transfer by Hardware Activation ..... IX - 37

Chapter10 SystemControl

10-1 Summery of Adress Break Function ..... X - 2
10-1-1 Adress Break ..... X - 2
10-1-2 Control Registers ..... X - 3
10-1-3 Adress Break Setup Example ..... X - 4
10-2 System-related Register Protection ..... X - 6
10-2-1 Overview ..... X - 6
10-2-2 System Control Register ..... X - 6

Chapter11 Ports

11-1 Summery of Ports ..... XI - 2
11-1-1 Overview ..... XI - 2
11-2 Control Registers ..... XI - 3
11-2-1 List of Port Control Registers ..... XI - 4
11-2-2 Register of Port 0 ..... XI - 6
11-2-3 Register of Port 1 ..... XI - 8
11-2-4 Register of Port 2 ..... XI - 10
11-2-5 Register of Port 3 ..... XI - 12
11-2-6 Register of Port 4 ..... XI - 14
11-2-7 Register of Port 5 ..... XI - 17
11-2-8 Register of Port 6 ..... XI - 19
11-2-9 Register of Port 7 ..... XI - 21
11-2-10 Register of Port 8 ..... XI - 23
11-2-11 Register of Port 9 ..... XI - 25
11-2-12 Register of Port A ..... XI - 27
11-2-13 Register of Port B ..... XI - 29
11-3 Port Block Diagram ..... XI - 30
11-4 Port Setup Example ..... XI - 41
11-4-1 General-purpose Port Setup ..... XI - 41

Chapter12 Appendix

12-1 Electrical Characteristics ..... XII - 2
12-2 MN102H74G/74F/74D/F74G Register Adress Map ..... XII - 26
12-3 List of Pin Functions ..... XII - 27
12-4 Initialization Program ..... XII - 29
12-5 Flash EEPROM Version ..... XII - 30
12-5-1 Overview ..... XII - 30
12-5-2 Flash EEPROM Programming ..... XII - 32
12-5-3 PROM Writer Mode ..... XII - 33
12-5-4 Onboard Serial Programming Mode ..... XII - 34
12-5-5 Connecting Onbord Serial Programming Mode ..... XII - 35

*Instruction Set (MN102H SERIES INSTRUCTION SET)
*Instruction Map (MN102H SERIES INSTRUCTION MAP)

Figures

Figure 1-1-1 Processor Status Word (PSW) ..... I - 5
Figure 1-1-2 Address Space (memory Expansion Mode) ..... I - 7
Figure 1-1-3 Interrupt Controller Configuration ..... I - 8
Figure 1-1-4 Interrupt Process Sequence ..... I - 8
Figure 1-3-1 Block Diagram ..... I - 10
Figure 1-4-1 Pin Configuration in Single-chip Mode ..... I - 12
Figure 1-4-2 Pin Configuration in Memory Expansion Mode ..... I - 13
Figure 1-4-3 Pin Configuration in Processor Mode ..... I - 14
Figure 1-4-4 OSCI and OSCO Connection Example ..... I - 27
Figure 1-4-5 XI and XO Connection Example ..... I - 27
Figure 1-4-6 Reset Pin Connection Example ..... I - 27
Figure 1-4-7 External Dimensions: 100-pin LQFP ..... I - 28

Figure 2-1-1 Machine Clock (No Wait) ..... II - 2
Figure 2-1-2 Machine Clock (Wait) ..... II - 2
Figure 2-1-3 Clock Signal at Half Wait ..... II - 3
Figure 2-1-4 Clock Siganl at Half Wait (C-Bus Access) ..... II - 3
Figure 2-2-1 Instruction Execution Controller ..... II - 4
Figure 2-2-2 Pipeline Process Example 1 ..... II - 5
Figure 2-2-3 Pipeline Process Example 2 ..... II - 6
Figure 2-3-1 Address Register ..... II - 7
Figure 2-3-2 Data Register ..... II - 8
Figure 2-3-3 Processor Status Word ..... II - 10
Figure 2-5-1 Transition between Operation Modes ..... II - 14
Figure 2-5-2 Operation Mode Control and Clock Oscillation ..... II - 16
Figure 2-5-3 Sequence of Switching to/from Standby Mode ..... II - 19
Figure 2-7-1 Sequence of Reset Release ..... II - 23
Figure 2-8-1 CPU Mode Control Register (CPUM) ..... II - 25
Figure 2-8-2 Extension Function Control Register (EFCR) ..... II - 26

Figure 3-1-1 Address Space (Processor Mode) ..... III - 2
Figure 3-1-2 Address Space (Memory Expansion Mode) ..... III - 3
Figure 3-2-1 Address Space ..... III - 4
Figure 3-2-2 External Bus Access Signal ..... III - 5
Figure 3-2-3 External Memory Mode Register 0 and 0S ..... III - 7
Figure 3-2-4 External Memory Mode Register 1 and 1S ..... III - 8
Figure 3-2-5 External Memory Mode Register 2 and 2S ..... III - 9
Figure 3-2-6 External Memory Mode Register 3 ..... III - 10
Figure 3-2-7 External Memory Mode Register 3S ..... III - 11

Figure 3-3-1 ROM Connection (16-bit Bus Width) ..... III - 13
Figure 3-3-2 ROM Connection (8-bit Bus Width) ..... III - 14
Figure 3-3-3 RAM Connection (16-bit Bus Width) ..... III - 15
Figure 3-3-4 RAM Connection (8-bit Bus Width) ..... III - 16
Figure 3-3-5 Byte Data Control Mode Access Timing ..... III - 16
Figure 3-3-6 C-Bus Access Timing ..... III - 17
Figure 3-3-7 16-bit Bus Access Timing (No Wait) ..... III - 18
Figure 3-3-8 16-bit Bus Access Timing (1 Wait) ..... III - 18
Figure 3-3-9 16-bit Bus Access Timing (1 Wait)(Late Access Mode) ..... III - 19
Figure 3-3-10 16-bit Bus Access Timing (1 Wait)(WE Short Mode) ..... III - 19
Figure 3-3-11 8-bit Bus Access Timing (No Wait) ..... III - 20
Figure 3-3-12 8-bit Bus Access Timing (1 Wait) ..... III - 20
Figure 3-3-13 8-bit Bus Access Timing (1 Wait)(Late Access Mode) ..... III - 21
Figure 3-3-14 8-bit Bus Access Timing (1 Wait)(WE Short Mode) ..... III - 21
Figure 3-3-15 16-bit Bus Handshake Access Timing ..... III - 22
Figure 3-3-16 8-bit Bus Handshake Access Timing ..... III - 22
Figure 3-3-17 Handshake Access Timing at Half Wait ..... III - 23
Figure 3-3-18 Wait Mask Function ..... III - 23
Figure 3-4-1 Bus Arbitration Timing ..... III - 25

Figure 4-1-1 Interrupt Control Configuration ..... IV - 4
Figure 4-1-2 Interrupt Group ..... IV - 5
Figure 4-1-3 Interrupt ServiceRoutine Sequence ..... IV - 6
Figure 4-1-4 Maskable Interrupt Control Register (GnICR) ..... IV - 8
Figure 4-1-5 Nonmaskable Interrupt Control Register (G0ICR) ..... IV - 9
Figure 4-1-6 Interrupt Acceptance Group Number Register (IAGR) ..... IV - 10
Figure 4-1-7 Subgroup Maskable Interrupt Control Register (SG50ICR) ..... IV - 11
Figure 4-1-8 Interrupt Acceptance Ditermination ..... IV - 12
Figure 4-1-9 Maskable Interrupt Process Sequence (without Multiple Interrupts) ..... IV - 14
Figure 4-1-10 External Interrupt Edge Specification 0 (EXTMD0) ..... IV - 15
Figure 4-1-11 External Interrupt Edge Specification 1 (EXTMD1) ..... IV - 16
Figure 4-1-12 Stack Pointer (SP) Operation at Interrupt ..... IV - 18
Figure 4-1-13 External InterruptSetup Timing ..... IV - 22
Figure 4-1-14 Watchdog Timer Interrupt Setup Timing ..... IV - 24
Figure 4-1-15 Stack Pointer (SP) Operation at Interrupt Return ..... IV - 28
Figure 4-1-16 Maskable Interrupt Process Sequence (with Multiple Interrupts) ..... IV - 30
Figure 4-2-1 Interrupt Accept Group Number Register (IAGR) ..... IV - 31
Figure 4-2-2 Nonmaskable Interrupt Control Register (G0ICR) ..... IV - 32
Figure 4-2-3 Extended Watchdog Interrupt Control Register (WDREG) ..... IV - 32

Figure 4-2-4 Maskable Interrupt Control Register (Group1)(G1ICR) ..... IV - 33
Figure 4-2-5 Maskable Interrupt Control Register (Group2)(G2ICR) ..... IV - 34
Figure 4-2-6 Maskable Interrupt Control Register (Group3)(G3ICR) ..... IV - 35
Figure 4-2-7 Maskable Interrupt Control Register (Group4)(G4ICR) ..... IV - 36
Figure 4-2-8 Maskable Interrupt Control Register (Group5)(G5ICR) ..... IV - 37
Figure 4-2-9 Maskable Interrupt Control Register (Group6)(G6ICR) ..... IV - 38
Figure 4-2-10 Maskable Interrupt Control Register (Group7)(G7ICR) ..... IV - 39
Figure 4-2-11 Maskable Interrupt Control Register (Group8)(G8ICR) ..... IV - 40
Figure 4-2-12 Maskable Interrupt Control Register (Group9)(G9ICR) ..... IV - 41
Figure 4-2-13 Subgroup Maskable Interrupt Control Register 40 (SG40ICR) ..... IV - 42
Figure 4-2-14 Subgroup Maskable Interrupt Control Register 50(SG50ICR) ..... IV - 43
Figure 4-2-15 Subgroup Maskable Interrupt Control Register 60(SG60ICR) ..... IV - 44
Figure 4-2-16 Subgroup Maskable Interrupt Control Register 61(SG61ICR) ..... IV - 45
Figure 4-2-17 Subgroup Maskable Interrupt Control Register 70(SG70ICR) ..... IV - 46
Figure 4-2-18 Subgroup Maskable Interrupt Control Register 71(SG71ICR) ..... IV - 47
Figure 4-2-19 Subgroup Maskable Interrupt Control Register 80(SG80ICR) ..... IV - 48
Figure 4-2-20 Subgroup Maskable Interrupt Control Register 81(SG81ICR) ..... IV - 49
Figure 4-2-21 Subgroup Maskable Interrupt Control Register 90(SG90ICR) ..... IV - 50
Figure 4-2-22 Subgroup Maskable Interrupt Control Register 91(SG91ICR) ..... IV - 51

Figure 5-1-1 Event Counter Timing (Timer 0 and Timer 1) ..... V - 3
Figure 5-1-2 Timer Output, Interval Timer Timing (Timer 0 and Timer 1) ..... V - 3
Figure 5-1-3 Relationship between Prescaler, Timer 0 and Timer 1 ..... V - 4
Figure 5-1-4 System Configuration ..... V - 4
Figure 5-1-5 Timer 0 Block Diagram ..... V - 5
Figure 5-1-6 Timer 1 Block Diagram ..... V - 5
Figure 5-1-7 Prescaler 0 Block Diagram ..... V - 6
Figure 5-1-8 8-bit Timer Block Diagram (Timer 2 to Timer 9) ..... V - 7
Figure 5-1-9 8-bit Timer Connection (Timer 2 to Timer 5) ..... V - 8
Figure 5-1-10 8-bit Timer Connection (Timer 6 to Timer 9) ..... V - 9
Figure 5-2-1 Timer 0 Base Register (TM0BR) ..... V - 11
Figure 5-2-2 Timer 1 Base Register (TM1BR) ..... V - 11
Figure 5-2-3 Prescaler 0 Base Register (PS0BR) ..... V - 11
Figure 5-2-4 Timer 0 Binary Counter (TM0BC) ..... V - 12
Figure 5-2-5 Timer 1 Binary Counter (TM1BC) ..... V - 12
Figure 5-2-6 Prescaler 0 Binary Counter (PS0BC) ..... V - 12
Figure 5-2-7 Timer 0 Mode Register (TM0MD) ..... V - 13
Figure 5-2-8 Timer 1 Mode Register (TM1MD) ..... V - 13
Figure 5-2-9 Prescaler 0 Mode Register (PS0MD) ..... V - 14

Figure 5-2-10 Timer 2 Base Register (TM2BR) ..... V - 16
Figure 5-2-11 Timer 3 Base Register (TM3BR) ..... V - 16
Figure 5-2-12 Timer 4 Base Register (TM4BR) ..... V - 16
Figure 5-2-13 Timer 5 Base Register (TM5BR) ..... V - 16
Figure 5-2-14 Timer 6 Base Register (TM6BR) ..... V - 17
Figure 5-2-15 Timer 7 Base Register (TM7BR) ..... V - 17
Figure 5-2-16 Timer 8 Base Register (TM8BR) ..... V - 17
Figure 5-2-17 Timer 9 Base Register (TM9BR) ..... V - 17
Figure 5-2-18 Timer 2 Binary Counter (TM2BC) ..... V - 18
Figure 5-2-19 Timer 3 Binary Counter (TM3BC) ..... V - 18
Figure 5-2-20 Timer 4 Binary Counter (TM4BC) ..... V - 18
Figure 5-2-21 Timer 5 Binary Counter (TM5BC) ..... V - 18
Figure 5-2-22 Timer 6 Binary Counter (TM6BC) ..... V - 19
Figure 5-2-23 Timer 7 Binary Counter (TM7BC) ..... V - 19
Figure 5-2-24 Timer 8 Binary Counter (TM8BC) ..... V - 19
Figure 5-2-25 Timer 9 Binary Counter (TM9BC) ..... V - 19
Figure 5-2-26 Timer 2 Mode Register (TM2MD) ..... V - 20
Figure 5-2-27 Timer 3 Mode Register (TM3MD) ..... V - 21
Figure 5-2-28 Timer 4 Mode Register (TM4MD) ..... V - 22
Figure 5-2-29 Timer 5 Mode Register (TM5MD) ..... V - 23
Figure 5-2-30 Timer 6 Mode Register (TM6MD) ..... V - 24
Figure 5-2-31 Timer 7 Mode Register (TM7MD) ..... V - 25
Figure 5-2-32 Timer 8 Mode Register (TM8MD) ..... V - 26
Figure 5-2-33 Timer 9 Mode Register (TM9MD) ..... V - 27
Figure 5-2-34 Prescaler 25 Control Register (PSCNT25) ..... V - 28
Figure 5-2-35 Prescaler 69 Control Register (PSCNT69) ..... V - 28
Figure 5-3-1 Event Counter Timing ..... V - 30
Figure 5-3-2 Clock Output Configuration Example ..... V - 31
Figure 5-3-3 Clock Output Timing ..... V - 33
Figure 5-3-4 Interval Timer Configuration Example ..... V - 34
Figure 5-3-5 Interval Timer Timing ..... V - 37
Figure 5-4-1 Interval Timer Operation ..... V - 39
Figure 5-4-2 Interval Timer Operation (Clock Source = SYSCLCK) ..... V - 39
Figure 5-4-3 Interval Timer Operation (with Prescaler) ..... V - 40
Figure 5-5-1 Event Count Operation ..... V - 44
Figure 5-6-1 Cascade Connection ..... V - 47
Figure 5-6-2 Timer 2, Timer 3 Operation 1 ..... V - 50
Figure 5-6-3 Timer 2, Timer 3 Operation 2 ..... V - 51
Figure 5-7-1 16-bit Timer Block Diagram ..... V - 55

Figure 5-7-2 16-bit Timer Connection ..... V - 56
Figure 5-8-1 Timer Compare/Capture A Register (TMnCA) ..... V - 58
Figure 5-8-2 Timer Compare/Capture B Register (TMnCB) ..... V - 59
Figure 5-8-3 Timer Compare/Capture Register Block Diagram ..... V - 60
Figure 5-8-4 Timer Binary Counter (TMnBC) ..... V - 61
Figure 5-8-5 Prescaler Control Register (PSCNTn) ..... V - 62
Figure 5-8-6 Timer 10 Mode Register (TM10MD) ..... V - 63
Figure 5-8-7 Timer 11 Mode Register (TM11MD) ..... V - 64
Figure 5-8-8 Timer 12 Mode Register (TM12MD) ..... V - 65
Figure 5-8-9 Timer 13 Mode Register (TM13MD) ..... V - 66
Figure 5-8-10 Timer Compare/Capture A Mode Register (TMnMDA) ..... V - 67
Figure 5-8-11 Timer Compare/Capture B Mode Register (TMnMDB) ..... V - 68
Figure 5-9-1 Timer Operation Example ..... V - 70
Figure 5-9-2 Up/Down Counting Select (TMnUD[1:0]='10') ..... V - 71
Figure 5-9-3 Up/Down Counting Select (TMnUD[1:0]='11') ..... V - 71
Figure 5-9-4 Count Operation Controlled by TMnIOA Pin ..... V - 72
Figure 5-9-5 Operation Example 1 ..... V - 73
Figure 5-9-6 Operation Example 2 ..... V - 73
Figure 5-9-7 Operation Example 3 ..... V - 74
Figure 5-9-8 Operation Example 4 ..... V - 74
Figure 5-9-9 Operation Example 5 ..... V - 75
Figure 5-9-10 Operation Example 6 ..... V - 76
Figure 5-9-11 Pin Output Waveform 1 ..... V - 81
Figure 5-9-12 Pin Output Waveform 2 ..... V - 81
Figure 5-9-13 Pin Output Waveform 3 ..... V - 82
Figure 5-9-14 Pin Output Waveform 4 ..... V - 82
Figure 5-9-15 At Upcount ..... V - 83
Figure 5-9-16 At Downcount ..... V - 83
Figure 5-9-17 Compare Capture A (B) Interrupt Request 1 ..... V - 84
Figure 5-9-18 Compare Capture A (B) Interrupt Request 2 ..... V - 84
Figure 5-9-19 Interval Timer Operation 1 ..... V - 87
Figure 5-9-20 Interval Timer Operation 2 ..... V - 87
Figure 5-9-21 Interval Timer Operation (Count Source = SYSCLK) ..... V - 88
Figure 5-9-22 Interval Timer Operation (Prescaler Used) ..... V - 88

Figure 6-1-1 Serial Interface Block Diagram ..... VI - 3
Figure 6-2-1 Serial Reception Register (SCAnRB) ..... VI - 5
Figure 6-2-2 Serial Transmission Register (SCAnTB) ..... VI - 5
Figure 6-2-3 Serial Control Register (SCAnCTR) ..... VI - 6

Figure 6-2-4 Serial Status Register (SCAnSTR) ..... VI - 7
Figure 6-2-5 Serail Status 1 Register (SCAnSTR1) ..... VI - 8
Figure 6-3-1 Asynchronous Simplex Mode ..... VI - 9
Figure 6-3-2 Asynchronous Duplex Mode ..... VI - 9
Figure 6-3-3 Synchronous Simplex Mode ..... VI - 9
Figure 6-3-4 Synchronous Duplex Mode ..... VI - 9
Figure 6-3-5 I^2C Mode Connection ..... VI - 9
Figure 6-3-6 Asynchronous Transmission Timing ..... VI - 11
Figure 6-3-7 Asynchronous Reception Timing ..... VI - 11
Figure 6-3-8 Synchronous Transmission Timing ..... VI - 12
Figure 6-3-9 Synchronous Reception Timing ..... VI - 12
Figure 6-3-10 Asynchronous Transmission Configuration ..... VI - 13
Figure 6-3-11 Asynchronous Bit Transmission Timing ..... VI - 17
Figure 6-3-12 Master Transmission Timing (with ACK) ..... VI - 20
Figure 6-3-13 Master Reception Timing (with ACK) ..... VI - 22

Figure 7-1-1 Analog Interface Configuration ..... VII - 3
Figure 7-2-1 A/D Conversion Control Register (ANCTR) ..... VII - 5
Figure 7-2-2 A/D0 Conversion Data Buffer (AN0BUF) ..... VII - 6
Figure 7-2-3 A/D1 Conversion Data Buffer (AN1BUF) ..... VII - 6
Figure 7-2-4 A/D2 Conversion Data Buffer (AN2BUF) ..... VII - 6
Figure 7-2-5 A/D3 Conversion Data Buffer (AN3BUF) ..... VII - 7
Figure 7-2-6 A/D4 Conversion Data Buffer (AN4BUF) ..... VII - 7
Figure 7-2-7 A/D5 Conversion Data Buffer (AN5BUF) ..... VII - 7
Figure 7-2-8 A/D6 Conversion Data Buffer (AN6BUF) ..... VII - 8
Figure 7-2-9 A/D7 Conversion Data Buffer (AN7BUF) ..... VII - 8
Figure 7-3-1 A/D Conversion Timing (S/H = 1cycle, 10-bit Resolution) ..... VII - 9
Figure 7-3-2 A/D Conversion Timing (S/H = 4cycle, 10-bit Resolution) ..... VII - 9
Figure 7-3-3 Single Channel/Single Conversion Timing ..... VII - 10
Figure 7-3-4 Multiple Channels/Single Conversion Timing ..... VII - 10
Figure 7-3-5 Single Channel/Continuous Conversion Timing ..... VII - 11
Figure 7-3-6 Multiple Channels/Continuous Conversion Timing ..... VII - 11
Figure 7-3-7 Single Channel A/D Conversion ..... VII - 14
Figure 7-3-8 Multiple Channel A/D Conversion ..... VII - 16
Figure 7-3-9 Multiple Channel A/D Conversion Timing ..... VII - 17
Figure 7-3-10 A/D Conversion Flow ..... VII - 17

Figure 8-3-1 USB Block Diagram ..... VIII - 4
Figure 8-4-1 Second Packet Write during First Packet Transfer ..... VIII - 5

Figure 8-4-2 Third Packet Write during Second Packet Transfer ..... VIII - 5
Figure 8-5-1 ATC Interface ..... VIII - 8
Figure 8-5-2 ATC Transfer ..... VIII - 9
Figure 8-5-3 ATC Transfer Operation for IN Endpoints (Dual Address Mode) ..... VIII - 10
Figure 8-5-4 ATC Transfer Operation for IN Endpoints (Single Address Mode) ..... VIII - 10
Figure 8-5-5 ATC Transfer Operation for OUT Endpoints (Dual Address Mode) ..... VIII - 11
Figure 8-5-6 ATC Transfer Operation for OUT Endpoints (Single Address Mode) ... VIII - 11
Figure 8-5-7 ATC Transfer Flow for IN Endpoints ..... VIII - 12
Figure 8-5-8 ATC Transfer Flow for OUT Endpoints ..... VIII - 13
Figure 8-6-1 Interrupt Signal ..... VIII - 16
Figure 8-6-2 Interrupt at Remaining Interrupt Triggers 1 ..... VIII - 16
Figure 8-6-3 Interrupt at Remaining Interrupt Triggers 2 ..... VIII - 17
Figure 8-6-4 Interrupt Acknowledgement and Clearing ..... VIII - 18
Figure 8-6-5 Multiple Endpoint Request within the Same Interrupt ..... VIII - 19
Figure 8-6-6 Example Interrupt Source Checking Flow ..... VIII - 19
Figure 8-6-7 Suspend Sequence ..... VIII - 24
Figure 8-8-1 Function Address Register ..... VIII - 31
Figure 8-8-2 Power Management Register ..... VIII - 32
Figure 8-8-3 ENDPI Register ..... VIII - 33
Figure 8-8-4 USBI Register ..... VIII - 35
Figure 8-8-5 ENDPIE Register ..... VIII - 36
Figure 8-8-6 USBIE Register ..... VIII - 37
Figure 8-8-7 Index Register ..... VIII - 38
Figure 8-8-8 Frame_Number 1 Register ..... VIII - 39
Figure 8-8-9 Frame_Number 2 Register ..... VIII - 39
Figure 8-8-10 ICSR1 Register ..... VIII - 40
Figure 8-8-11 ICSR2 Register ..... VIII - 43
Figure 8-8-12 ICSR3 Register ..... VIII - 44
Figure 8-8-13 OCSR1 Register ..... VIII - 45
Figure 8-8-14 OCSR2 Register ..... VIII - 47
Figure 8-8-15 EP0CSR Register ..... VIII - 48
Figure 8-8-16 EP0CSRX Register ..... VIII - 50
Figure 8-8-17 MAXP Register ..... VIII - 51
Figure 8-8-18 OWC1 Register ..... VIII - 52
Figure 8-8-19 OWC2 Register ..... VIII - 52
Figure 8-8-20 Sampling Buffer EP3 ..... VIII - 53
Figure 8-8-21 Sampling Buffer EP4 ..... VIII - 53
Figure 8-8-22 ATC_SWITCH Register ..... VIII - 54

Figure 9-1-1 ATC Block Diagram ..... IX - 3
Figure 9-2-1 ATC Control Register ..... IX - 5
Figure 9-2-2 ATC Memory Mode Register ..... IX - 8
Figure 9-2-3 ATC Interrupt Register ..... IX - 10
Figure 9-2-4 ATC Interrupt Signal ..... IX - 10
Figure 9-2-5 ATC Transfer Word Count Register ..... IX - 11
Figure 9-2-6 ATC Source Address Register ..... IX - 11
Figure 9-2-7 ATC Destination Address Register ..... IX - 12
Figure 9-2-8 ATC Transfer Channel Priority Setup Register ..... IX - 13
Figure 9-2-9 ATC Input/Output Channel Select Register ..... IX - 14
Figure 9-2-10 ATC Input/Output Channel Select Diagram ..... IX - 14
Figure 9-2-11 ATC Hardware Activation Factor Select Register ..... IX - 15
Figure 9-3-1 ATC Operation ..... IX - 16
Figure 9-3-2 Burst Transfer Mode Sequence ..... IX - 17
Figure 9-3-3 1 Word Transfer Mode Sequence ..... IX - 17
Figure 9-3-4 ATC Transfer Sequence for Single Channel ..... IX - 18
Figure 9-3-5 ATC Transfer Sequence for Multiple Channels ..... IX - 19
Figure 9-3-6 Dual Address Mode ..... IX - 20
Figure 9-3-7 Single Address Mode (Source Address Specification) ..... IX - 20
Figure 9-3-8 Single Address Mode (Destination Address Specification) ..... IX - 21
Figure 9-3-9 C-bus Mode Base Cycle (No Wait, 2 Cycles) ..... IX - 22
Figure 9-3-10 C-bus Mode Base Cycle (Read Synchronization) ..... IX - 22
Figure 9-3-11 Early Read/Early Write (No Wait, 2 Cycles) ..... IX - 23
Figure 9-3-12 1 Cycle Operation Mode ..... IX - 24
Figure 9-3-13 C-bus Mode (1 Wait, 3 Cycles) ..... IX - 25
Figure 9-3-14 Early Read/Early Write Mode (1 Wait, 3 Cycles) ..... IX - 25
Figure 9-3-15 1 Wait Insert by Wait Signal in Handshake Mode WAIT Mask Cycle: 0 Cycle . IX - 26
Figure 9-3-16 2 Cycles Insert by Wait Signal in Handshake Mode WAIT Mask Cycle: 1 Cycle IX - 26
Figure 9-3-17 Transfer Example 1 ..... IX - 27
Figure 9-3-18 Transfer Example 2 ..... IX - 28
Figure 9-3-19 Transfer Example 3 ..... IX - 29
Figure 9-3-20 Transfer Example 4 ..... IX - 31
Figure 9-3-21 External Bus Request Arbitration ..... IX - 34
Figure 9-3-22 Transfer Suspend Using NMI ..... IX - 35
Figure 9-4-1 Process Flow ..... IX - 37

Figure 10-1-1 Address Break Block Diagram ..... X - 2
Figure 10-1-2 Address Break Operation Example ..... X - 2
Figure 10-1-3 Address Break Control Register (ADBCTL) ..... X - 3

Figure 10-1-4 Address Break n Address Pointer (ADB0) ..... X - 3
Figure 10-1-5 Address Break Setup Flow ..... X - 4
Figure 10-1-6 Stack State after NMI Interrupt ..... X - 5
Figure 10-2-1 System Control Register (SYSCTL) ..... X - 6

Figure 11-1-1 I/O Port Configuration ..... XI - 2
Figure 11-4-1 General-purpose Port Setup Example ..... XI - 41
Figure 11-4-2 Basic Flowchart of General-purpose Port Input ..... XI - 42
Figure 11-4-3 Basic Flowchart of General-purpose Port Output ..... XI - 42

Figure 12-1-1 System Clock Timing ..... XII - 18
Figure 12-1-2 Reset Timing ..... XII - 18
Figure 12-1-3 Power Reset Timing ..... XII - 18
Figure 12-1-4 Data Transfer Signal Timing (0.5 Wait) ..... XII - 19
Figure 12-1-5 Data Transfer Signal Timing (1 or Larger Wait) ..... XII - 20
Figure 12-1-6 Data Transfer Signal Timing (Byte Data Control Mode) ..... XII - 21
Figure 12-1-7 ATC Data Transfer Signal Timing (C-bus Basic) ..... XII - 22
Figure 12-1-8 ATC Data Transfer Signal Timing (1 Cycle Operation Mode) ..... XII - 23
Figure 12-1-9 Bus Open Request Signal Timing ..... XII - 24
Figure 12-1-10 Interrupt Signal Input Timing ..... XII - 24
Figure 12-1-11 Serial Interface Signal Timing 1 ..... XII - 24
Figure 12-1-12 Serial Interface Signal Timing 2 ..... XII - 24
Figure 12-1-13 Serial Interface Signal Timing 3 ..... XII - 25
Figure 12-1-14 Timer/Counter Signal Timing ..... XII - 25
Figure 12-5-1 Flash EEPROM Memory Map ..... XII - 31
Figure 12-5-2 Flash EEPROM Program Flow ..... XII - 32
Figure 12-5-3 Pin Configuration During Serial Programming ..... XII - 35
Figure 12-5-4 10 Pins Flat Cable for Interface Unit ..... XII - 37

Tables

Table 1-1-1	Memory Mode	I - 7
Table 1-2-1	Basic Specification	I - 9
Table 1-3-1	Block Function	I - 11
Table 1-4-1	List of Pin Functions	I - 15
Table 1-5-1	Addressing Mode	I - 29
Table 1-6-1	List of Instructions	I - 30
Table 2-3-1	Interrupt Mask Level and Interrupt Acceptance	II - 12
Table 2-4-1	List of Special Function Registers	II - 13
Table 2-7-1	Initial Value of Register at Reset	II - 24
Table 3-2-1	List of Bus Interface Control Registers	III - 5
Table 3-2-2	External Bus Access Mode	III - 12
Table 4-1-1	Interrupt Controller Overview	IV - 2
Table 4-1-2	Mask Level and Interrupt Level	IV - 9
Table 5-1-1	List of 8-bit Timer Functions	V - 2
Table 5-2-1	List of Timer 0, Timer 1 and Prescaler 0 Control Registers	V - 10
Table 5-2-2	List of Timer 2 to Timer 9 Control Registers	V - 15
Table 5-4-1	List of Control Registers for Interval Timer and Timer Output	V - 41
Table 5-5-1	List of Control Registers for Event Counter	V - 45
Table 5-6-1	List of Control Registers for Cascade Connection	V - 52
Table 5-7-1	List of 16-bit Timer Functions	V - 54
Table 5-8-1	List of 16-bit Timer Control Registers	V - 57
Table 5-9-1	Counting Operation When 2-phase Encorder (×1) is Selected	V - 69
Table 5-9-2	Counting Operation When 2-phase Encorder (×4) is Selected	V - 69
Table 5-9-3	TMnIOA Input Level	V - 72
Table 5-9-4	Compare/Capture Operating Mode Select	V - 77
Table 5-9-5	TMnIOA Pin Polarity Select	V - 78
Table 5-9-6	TMnIOB Pin Polarity Select	V - 78
Table 5-9-7	Capture Operating Mode Select	V - 79
Table 5-9-8	TMnIOA Output Waveform Select	V - 80
Table 5-9-9	TMnIOB Output Waveform Select	V - 80
Table 6-1-1	Serial Interface 0, 1 Function	VI - 2
Table 6-1-2	Serial Interface 2, 3 Function	VI - 2
Table 6-2-1	List of Serial Interface Control Registers	VI - 4
Table 6-3-1	Baud Rate Setup in Asynchronous Mode	VI - 10

Table 7-1-1 A/D Converter Functions ..... VII - 2
Table 7-2-1 List of Analog Interface Control Registers ..... VII - 4
Table 7-3-1 A/D Conversion Input Pin Setup ..... VII - 12
Table 7-3-2 A/D Conversion Clock Setup ..... VII - 12
Table 7-3-3 A/D Conversion Sampling Time Setup ..... VII - 13
Table 7-3-4 A/D Ladder Resistor Setup ..... VII - 13
Table 7-3-5 List of A/D Conversion Control Registers for Setup (Single Channel) ... VII - 14
Table 7-3-6 List of A/D Conversion Control Registers for Setup (Multiple Channels) VII - 16

Table 8-1-1 Endpoint Specifications ..... VIII - 2
Table 8-6-1 Interrupt Triggers ..... VIII - 14
Table 8-8-1 List of USB Control Registers ..... VIII - 30

Table 9-1-1 ATC Functions ..... IX - 2
Table 9-2-1 List of ATC Control Registers ..... IX - 4
Table 9-2-2 ATC Memory Space ..... IX - 7
Table 9-2-3 Endpoint Select List in Single Address Mode ..... IX - 14
Table 9-2-4 ATRQSEL Register Setup ..... IX - 15

Table 11-2-1 List of Port Control Registers ..... XI - 4
Table 11-3-1 Port Block Diagram ..... XI - 30

Table 12-5-1 Pin Connection of MN102HF74G and Flash Writer ..... XII - 37

Chapter 1 General Description

1-1 General Description

1-1-1 Introduction

The 16-bit MN102 series high-speed linear addressing version designs the new architecture for C-language programming based on a detailed analysis for embedded applications. This improves the previous system architecture in speed and function to meet the requirements in user systems including miniaturization to power consumption.

This series adapts a load/store architecture method for computing within registers instead of the accumulator system for computing within the memory space in the previous series. The basic instructions are one byte/one machine cycle. This reduces code size and improves compiler efficiency. This series uses the circuit designed for submicron technology providing optimized hardware and low system power consumption.

This series has up to 16 MB of linear address space and can develop the highly efficient programs. The optimized hardware architecture allows lower power consumption even in large systems.

1-1-2 Features

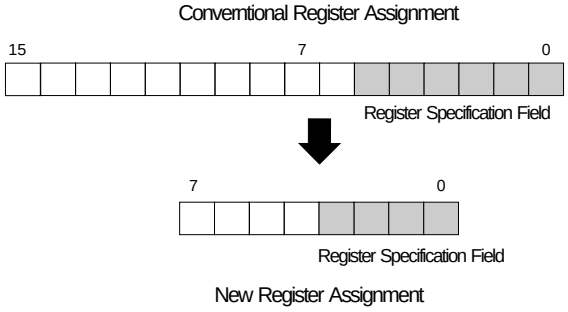
This series contains a flexible and optimized hardware architecture as well as a simple and efficient instruction set. This allows economy and speed. This section describes the features of this series CPU.

1. Linear Addressing for Large Systems

The MN102H series contains up to 16 MB of linear address space. The CPU does not detect borders between address spaces, which provides an effective development environment. The hardware architecture is also optimized for large systems. The memory is not divided into instruction areas and data areas so that operations can share instructions.

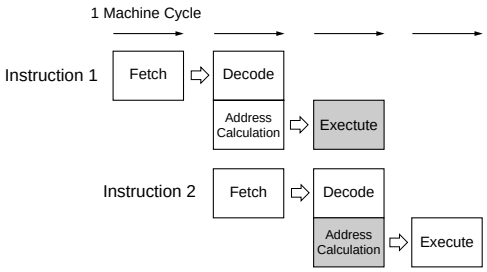
2. Single-byte Basic Instruction Length

The MN102H series has replaced general registers with eight internal CPU registers divided four address registers (A0-A3) and four data registers (D0-D3). The register specification fields are four bits or less, and the code size of the basic instructions including register-to register operations and load/store operations is one byte.



3. High-speed Pipeline Processing

The MN102H series executes instructions in a 3-stage pipeline: fetch, decode, execute. This allows the MN102H series to execute instructions of single byte in one machine cycle.

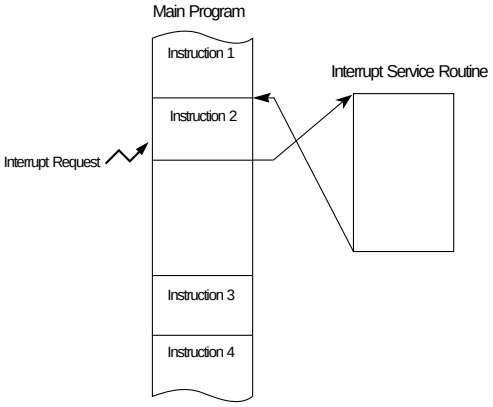


4. Simple Instruction Set

The MN102H series uses an instruction set of 41 instructions, designed specially for the programming model for embedded applications. To shrink code size, instructions have a variable length of 1 to 7 bytes. The most frequently used instructions in C-language compiler are single byte.

5. High-speed Interrupt Response

The MN102H series halts the instructions execution even during the execution of the instruction with long execution cycles. After an interrupt occurs, the program moves to the interrupt service routine within 6 cycles or less. The MN102H series improves real-time control performance using the interrupt handler which adjusts interrupt servicing speed.



6. Flexible Interrupt Control Structure

The interrupt controller contains 10 groups (of them, Group 0 is reserved for NMI interrupt) and supports a maximum of 21 interrupt vectors, assigning up to 3 vectors to groups. Each group can be set to one of seven priority levels. This provides the software design flexibility and control. The CPU is compatible with software from previous Panasonic peripheral modules.

7. High-speed, High-functional External Interface

The MN102H series supports external interface functions including DMA, handshake function and bus arbitration.

8. C-language Development Environment

The MN102H series has simple hardware optimized for C-language programming and highly efficient C compiler. With this advantage, this series improves development environment for C-language embedded applications without expanding the program size. The **PanaXSeries** development tools support the MN102H series devices.

9. Outstanding Power Savings

The MN102H series contains separate buses for instructions, data and peripheral functions, which distribute and reduce load capacitance. This reduces overall power consumption. The MN102H series also supports three modes of SLOW, HALT and STOP for power savings.

10. High-speed Signal Processing

An internal multiplier operates 16-bit × 16-bit = 32-bit in a single cycle. In addition, the hardware contains a saturation calculator which must be used in signal processing and increases the signal processing speed.

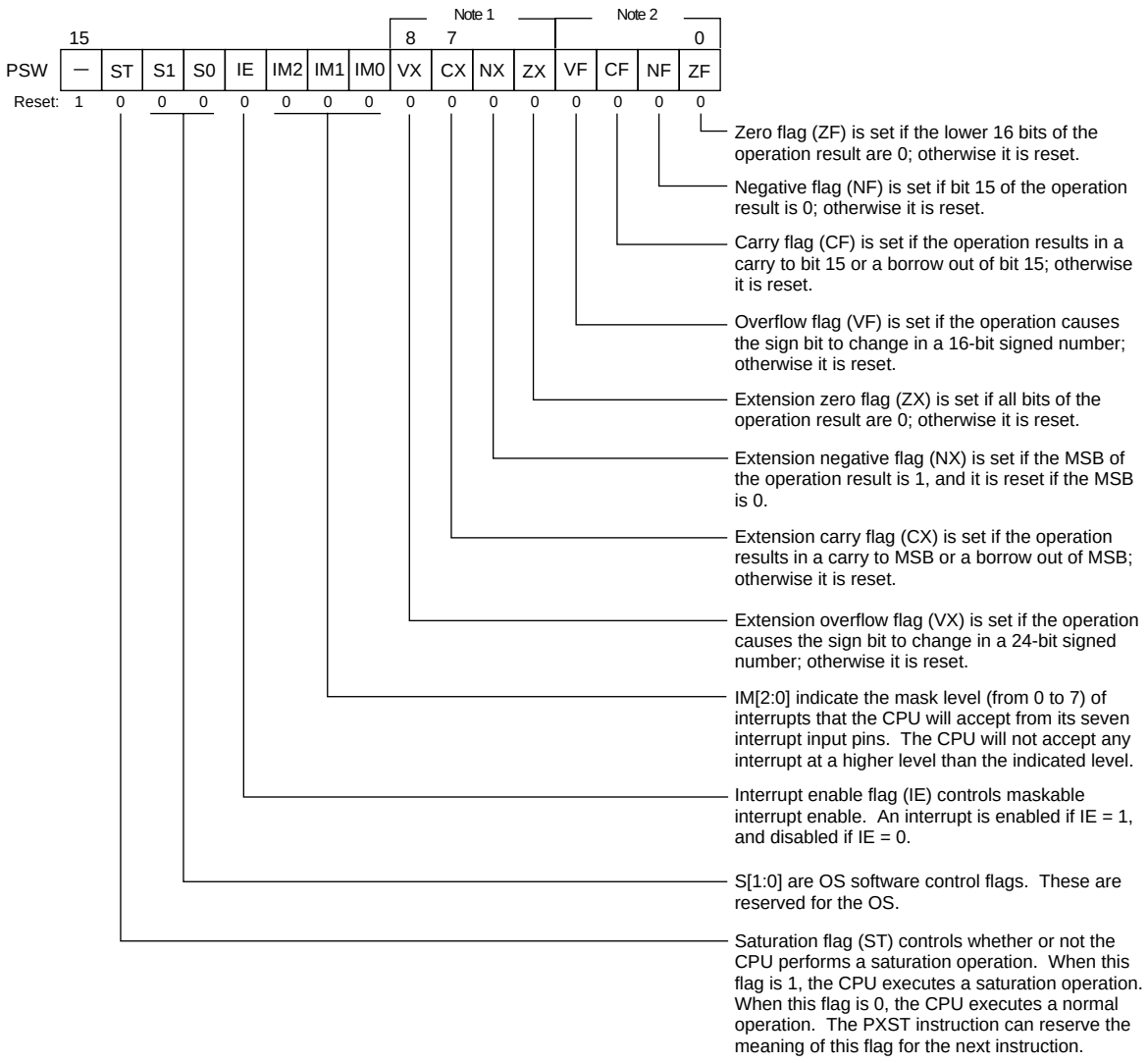
PanaXSeries is a trademark of the Matsushita Electric Industrial Co., Ltd.

1-1-3 Overview

This section describes the basic configuration and functions of the MN102H74G/74F/74D/F74G.

■ Processor Status Word (PSW)

The PSW register contains the operating result flags and interrupt mask level flags.


Note 1
Note 2
15
8
7
0
PSW
—
ST
S1
S0
IE
IM2
IM1
IM0
VX
CX
NX
ZX
VF
CF
NF
ZF
Reset: 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Zero flag (ZF) is set if the lower 16 bits of the operation result are 0; otherwise it is reset.
Negative flag (NF) is set if bit 15 of the operation result is 0; otherwise it is reset.
Carry flag (CF) is set if the operation results in a carry to bit 15 or a borrow out of bit 15; otherwise it is reset.
Overflow flag (VF) is set if the operation causes the sign bit to change in a 16-bit signed number; otherwise it is reset.
Extension zero flag (ZX) is set if all bits of the operation result are 0; otherwise it is reset.
Extension negative flag (NX) is set if the MSB of the operation result is 1, and it is reset if the MSB is 0.
Extension carry flag (CX) is set if the operation results in a carry to MSB or a borrow out of MSB; otherwise it is reset.
Extension overflow flag (VX) is set if the operation causes the sign bit to change in a 24-bit signed number; otherwise it is reset.
IM[2:0] indicate the mask level (from 0 to 7) of interrupts that the CPU will accept from its seven interrupt input pins. The CPU will not accept any interrupt at a higher level than the indicated level.
Interrupt enable flag (IE) controls maskable interrupt enable. An interrupt is enabled if IE = 1, and disabled if IE = 0.
S[1:0] are OS software control flags. These are reserved for the OS.
Saturation flag (ST) controls whether or not the CPU performs a saturation operation. When this flag is 1, the CPU executes a saturation operation. When this flag is 0, the CPU executes a normal operation. The PXST instruction can reserve the meaning of this flag for the next instruction.

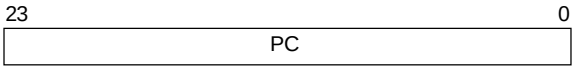
Note 1: These bits change depending on all 24 bits of the operation result.
Note 2: These bits change depending on the lower 16 bits of the operation result.
The IE flag should be set to 0 before the IM[2:0] flags of PSW are changed.

Figure 1-1-1 Processor Status Word (PSW)

Please refer to “Chapter 12 MN102H SERIES INSTRUCTION SET” for which flag each instruction should be issued.

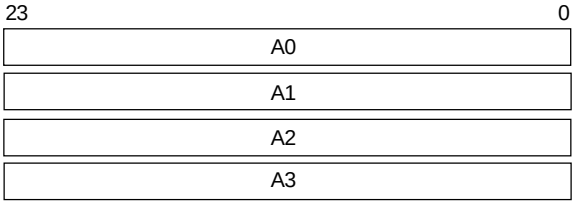
■ Internal Registers, Memory, and Special Function Registers

Program Counter



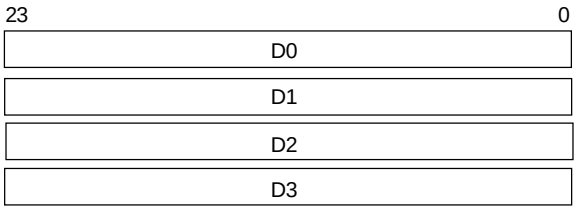
The program counter specifies the 24-bit address of the program during the execution.

Address Registers



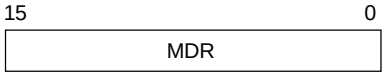
The address registers specify the data location on memory. Of four registers, A3 is assigned as the stack pointer.

Data Registers



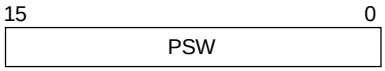
The data registers perform all arithmetic and logic operations. When the byte-length (8-bit) or word-length (16-bit) data is transferred to memory or to another register, an instruction adds a zero or sign extension.

Multiplication/Division Register



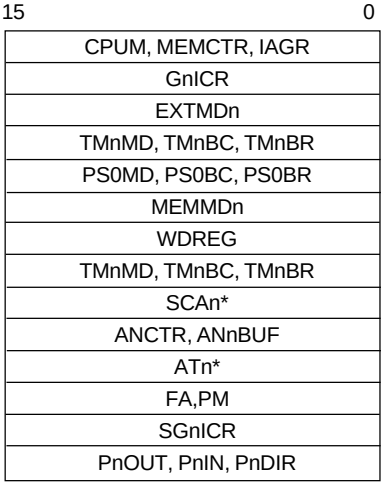
The multiplication/division register stores the upper 16 bits of the 32-bit product of multiplication operations. In division operations, this register stores the upper 16 bits of the 32-bit dividend before execution, and the 16-bit remainder of the quotient after execution.

Processor Status Word



Special function registers for peripheral function incorporated in the CPU Core are assigned to the I/O control register address space. Other registers for peripheral function are assigned to other address spaces.

Memory, Special Function Registers, I/O Ports

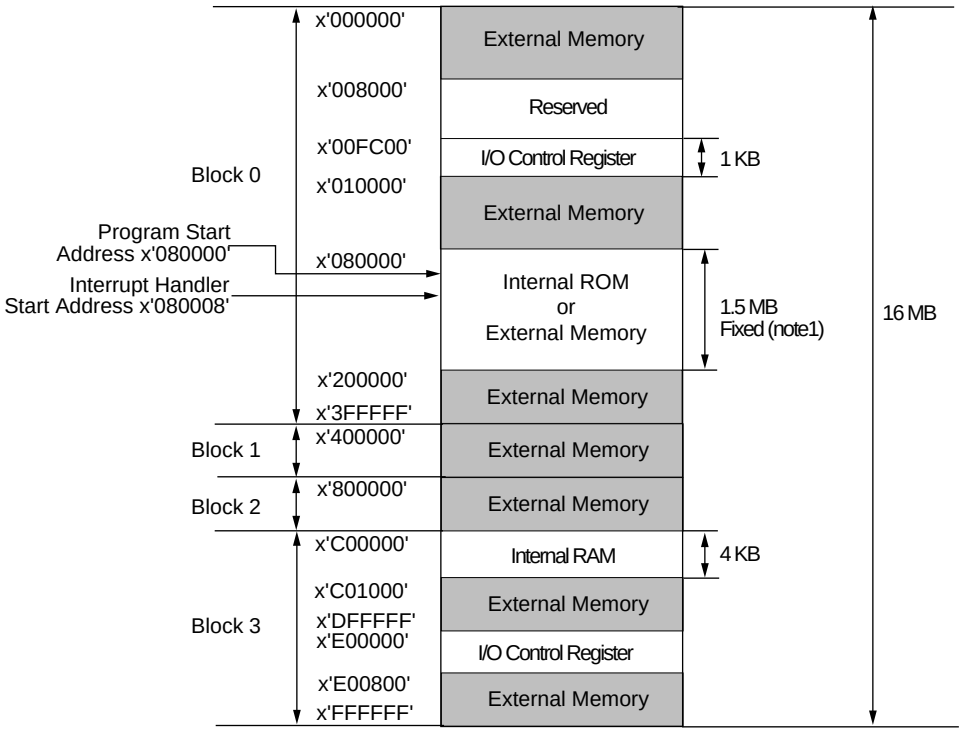


Internal Control Registers
Interrupt Control Registers
External Interrupt Edge Specification Registers
Timer/Counter Registers (n=0,1)
Prescaler Registers
Memory Control Registers
Watchdog Timer Interrupt Extension Control Registers
Timer/Counter Registers (n=2 to 13) *
Serial Interface Control Registers *
A/D Converter Registers *
ATC *
USB *
Extension Interrupt Control Registers *
I/O Port Registers *

* This allocation is an example. Actual memory, peripheral functions, special function registers and I/O port allocation depends on the model.

■ Address Space

The MN102H74G/74F/74D/F74G contains linear address space of up to 16 MB. There is no difference between instruction space and data space. Three memory modes are available depending on user program sizes.



Note1: The internal ROM space is fixed to 1.5 MB. The empty space is reserved.
Note2: Each ROM capacity depends on the model.

	ROM Capacity
MN102HF74G	128 KB
MN102H74G	128 KB
MN102H74F	96 KB
MN102H74D	64 KB

Figure 1-1-2 Address Space

Table 1-1-1 Memory Mode

Mode	Address Bit Width	ROM Capacity	External Memory Access
Single-chip Mode	-	64 KB /96 KB /128 KB	Disable
Memory Expansion Mode	Up to 24 Bits		Enable
Processor Mode		None	

Note: Address space x'008000' to x'00FBFF is reserved.

■ Interrupt Controller

Nonmaskable interrupt and maskable interrupts except reset are controlled by interrupt controller (Group 0 to Group 9) allocated in CPU externally. Each group has interrupt vectors and specifies one of 7 priority levels (priority level of interrupt acceptance).

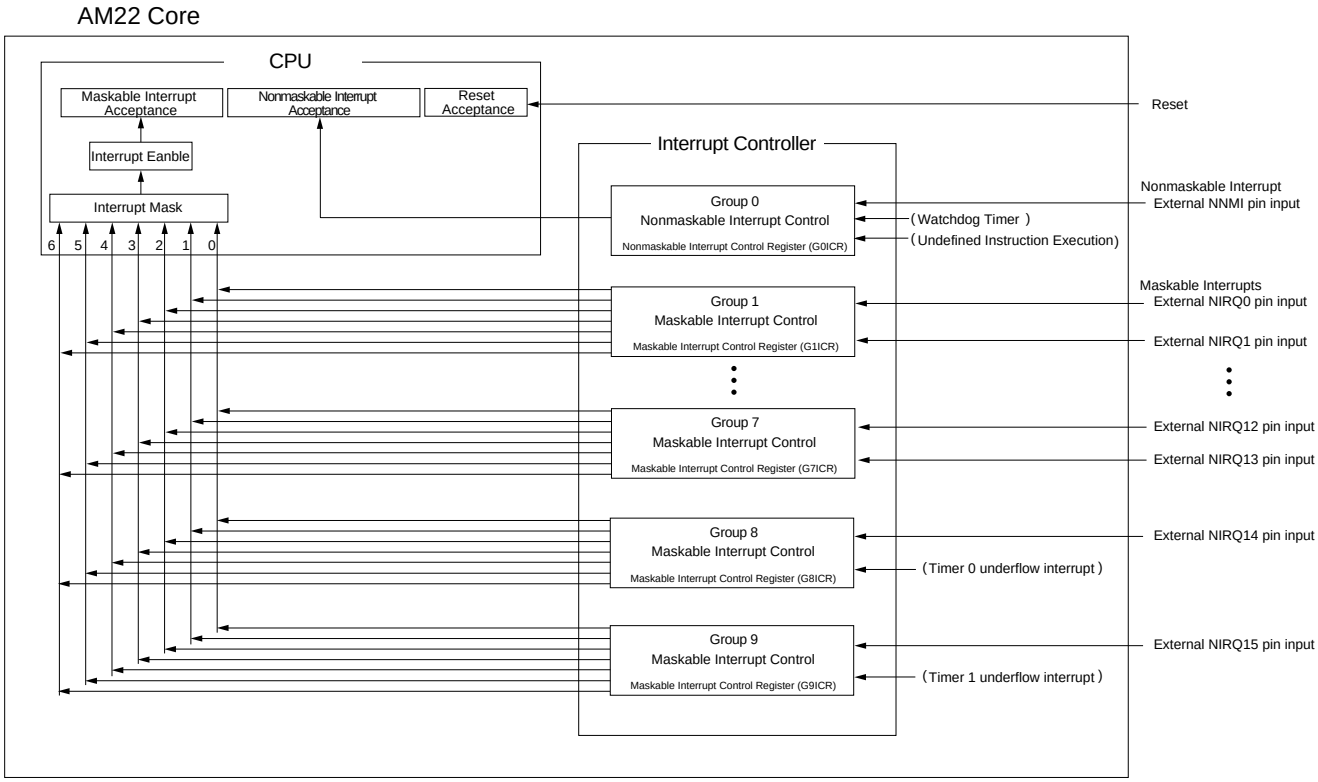


Figure 1-1-3 Interrupt Controller Configuration

The CPU checks the processor status word to determine whether an interrupt request is accepted or not. If an interrupt is accepted, automatic processing by hardware starts and the program and PSW are pushed to the stack. Next, the program moves to interrupt processing to search the interrupt vector and branch to the entry address of its corresponded interrupt process program.

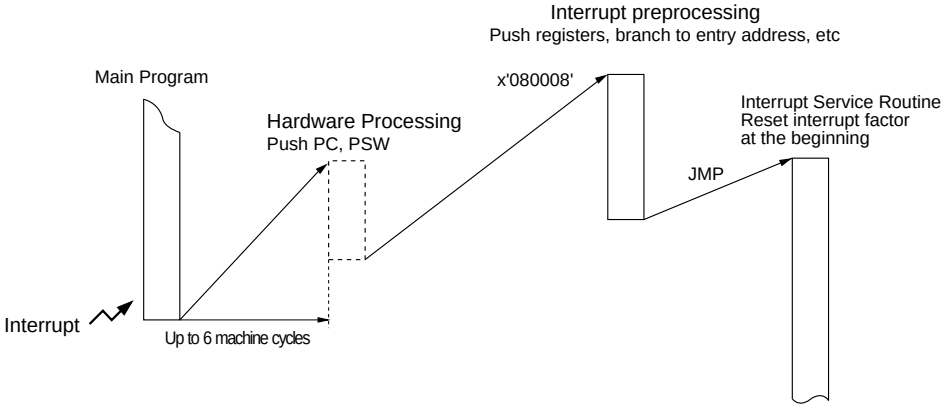


Figure 1-1-4 Interrupt Process Sequence

1-2 Basic Specification

This section describes basic specifications of AM22 Core.

Table 1-2-1 Basic Specification

Structure	Built-in multiplier (16 bits × 16 bits = 32 bits), Built-in saturate operation calculator Load/store architecture Eight registers: Four 24 bit data registers Four 24 bit address registers Others: 24 bit program counter 16 bit processor status word 16 bit multiplication/division register
Instruction	41 instructions 6 addressing modes 1-byte basic instruction length Code assignment: 1-byte (base) + 0 to 6 bytes (extension)
Performance	12 MHz internal operating frequency with a 12 MHz external oscillator Instruction execution clock cycles: For register-to-register operations, minimum 1 cycle (83 ns with a 12 MHz external oscillator) For branch operations, minimum 2 cycles (167 ns with a 12 MHz external oscillator) For load/store operations, minimum 1 cycle (83 ns with a 12 MHz external oscillator)
Pipeline	three stages: Instruction fetch, decode and execution
Address Space	Up to 16 MB linear address space Instruction and data shared space
External Bus	24 bit address 8/16 bit data Minimum bus cycle: 125 ns with a 12 MHz external oscillator Handshake mode or fixed wait mode
Interrupt	one external nonmaskable interrupt sixteen external interrupts two timer interrupts seven interrupt priority levels
Low-power Mode	SLOW mode, STOP mode, HALT mode
Timer	two 8 bit timers one 8 bit prescaler

1-3 Block Diagram

The following is the MN102H74G/74F/74D/F74G Block Diagram with AM22 core.

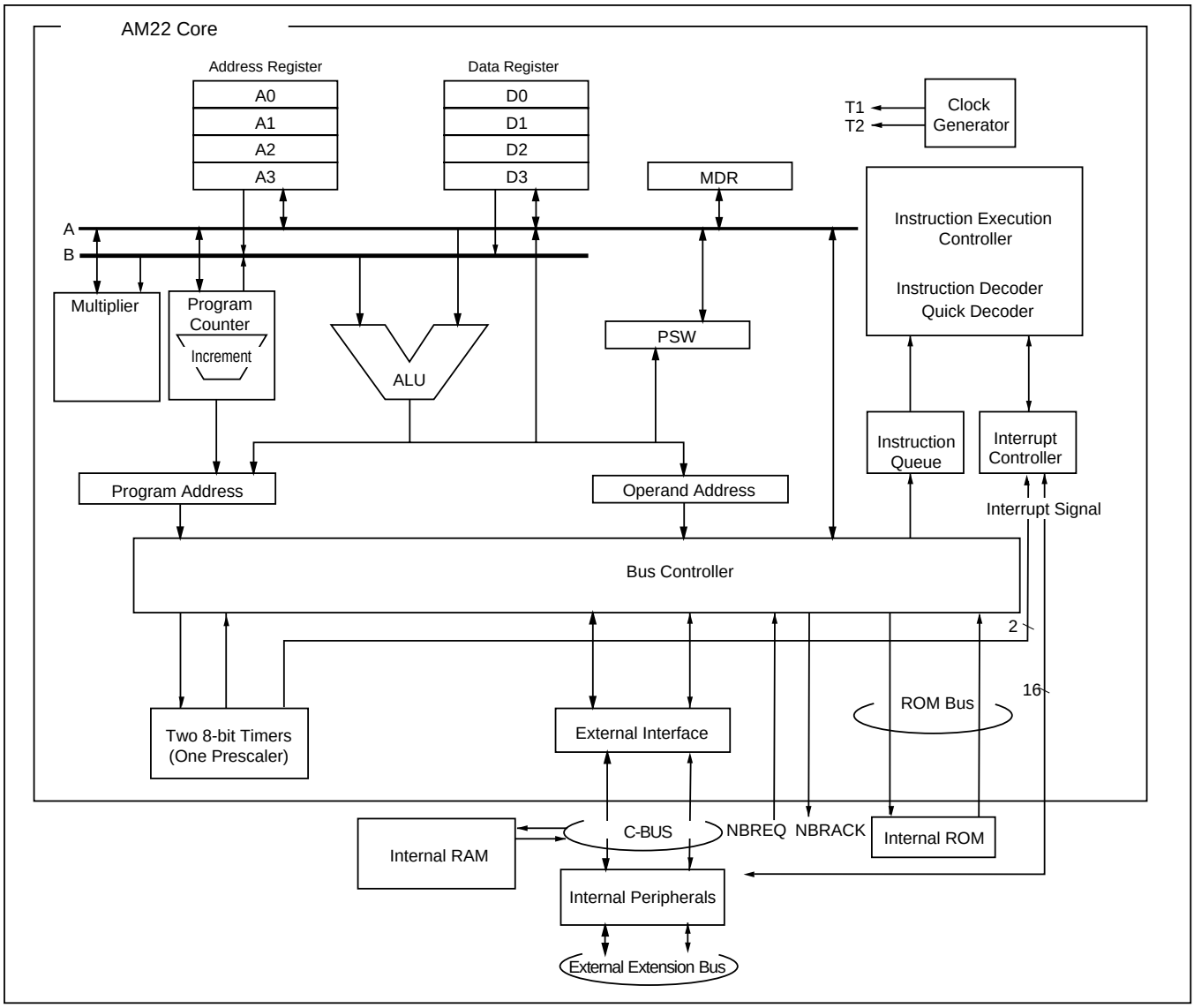


Figure 1-3-1 Block Diagram

Note: The external memory areas for AM22 core external interface are allocated as the external memory areas in MN102H74G/74F/74D/F74G.

Table 1-3-1 Block Function

Block	Function	Remark
Clock Generator	Clock Generator supplies the clock to all blocks in the CPU.	2-1
Program Counter	The program counter generates instruction addresses for instruction queues. Normally, the program counter increments based on the sequencer indication, but it sets the branch address or ALU operation results for the branch instruction and interrupt acceptance.	2-3
Instruction Queue	The instruction queue stores up to 4 bytes of the prefetched instruction.	2-2
Instruction Decoder	The instruction decoder decodes the instruction queue and generates control signals to execute the instruction and then executes the instruction by controlling each blocks.	
Quick Decoder	The quick decoder the 2-byte or larger instruction at faster speed.	
Instruction Execution Controller	Instruction execution controller controls each CPU block based on instruction decode results or interrupt requests.	
ALU	The ALU calculates operand addresses for arithmetic operations, logic operations and shift operations in register relative indirect addressing mode, indexed addressing mode, indexed addressing mode and register indirect addressing mode.	—
Address Registers (An)	The address register (An) stores the address on the memory for data transfer. The address register stores the base address in register relative indirect addressing mode, indexed addressing mode, indexed addressing mode and register indirect addressing mode.	2-3
Operation Registers (Dn, MDR)	The data register (Dn) stores results of arithmetic operations and data transferred to the memory. The address register stores the offset address in register relative indirect addressing mode, indexed addressing mode, indexed addressing mode and register indirect addressing mode. The MDR stores the results of multiplication/division operations.	
PSW	The PSW stores flags that indicate the status of the CPU interrupt controller and operation results.	
Interrupt Controller	The interrupt controller detects interrupt requests from 8-bit timers or peripheral functions and requests the CPU to move to the interrupt service routine.	4-1
Bus Controller	The bus controller controls the connection between the CPU internal bus and the CPU external bus. The bus controller has the bus arbitration function. The memory connected to AM22 core through external interface is treated as external memory.	3-1
Internal ROM,RAM	Internal ROM and internal RAM are allocated on program area, data area or stack area.	3-1
Peripherals	The peripheral functions are connected to AM22 core using C-bus.	—

1-4 Pin Function

1-4-1 Single-chip Mode

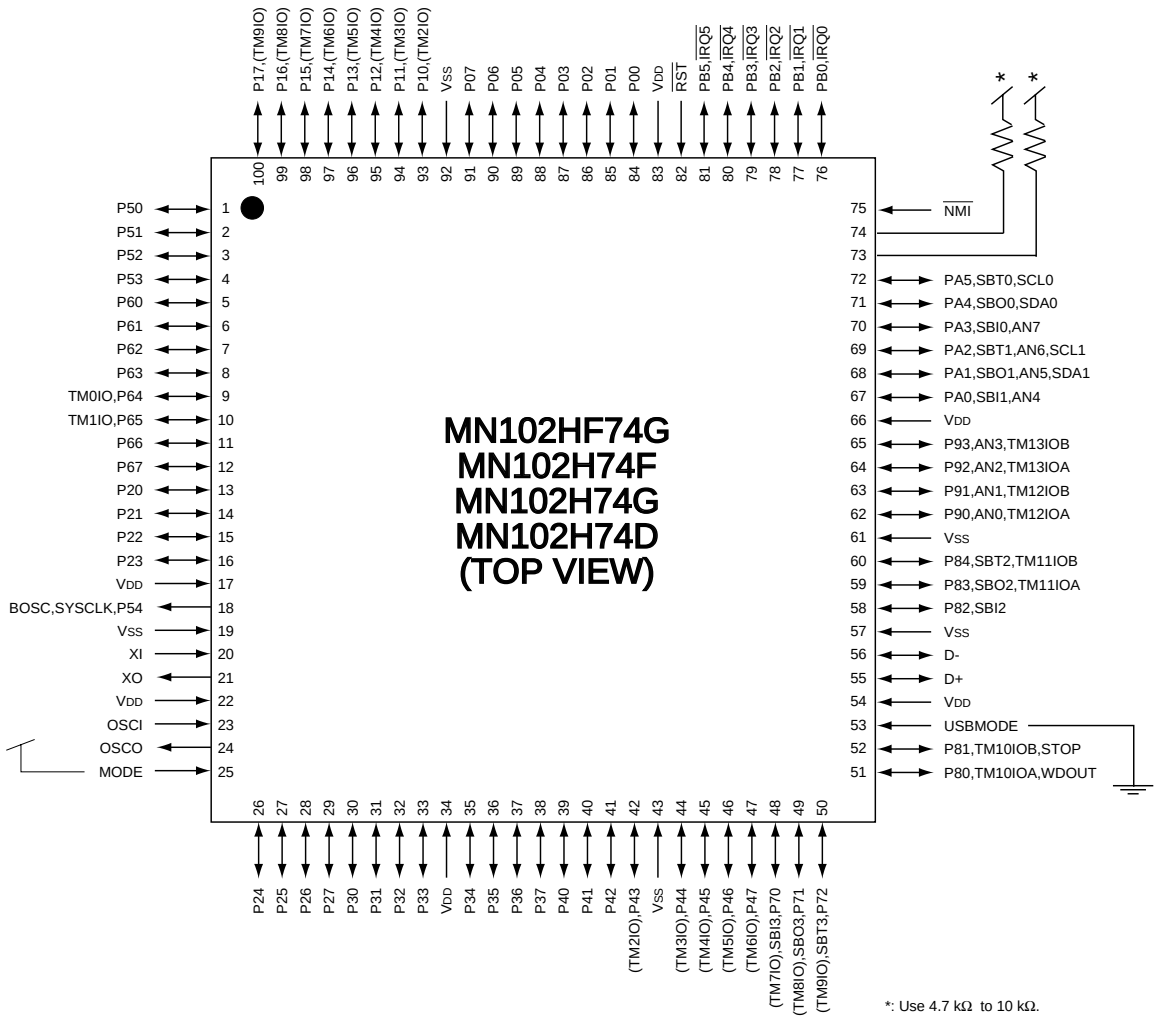


Figure 1-4-1 Pin Configuration in Single-chip Mode

> Unused pins require handling in the circuit (input pins are connected to VDD/VSS, output pins are open and input/output pins are connected to VDD/VSS or leave open depending on pin direction).

1-4-2 Memory Expansion Mode

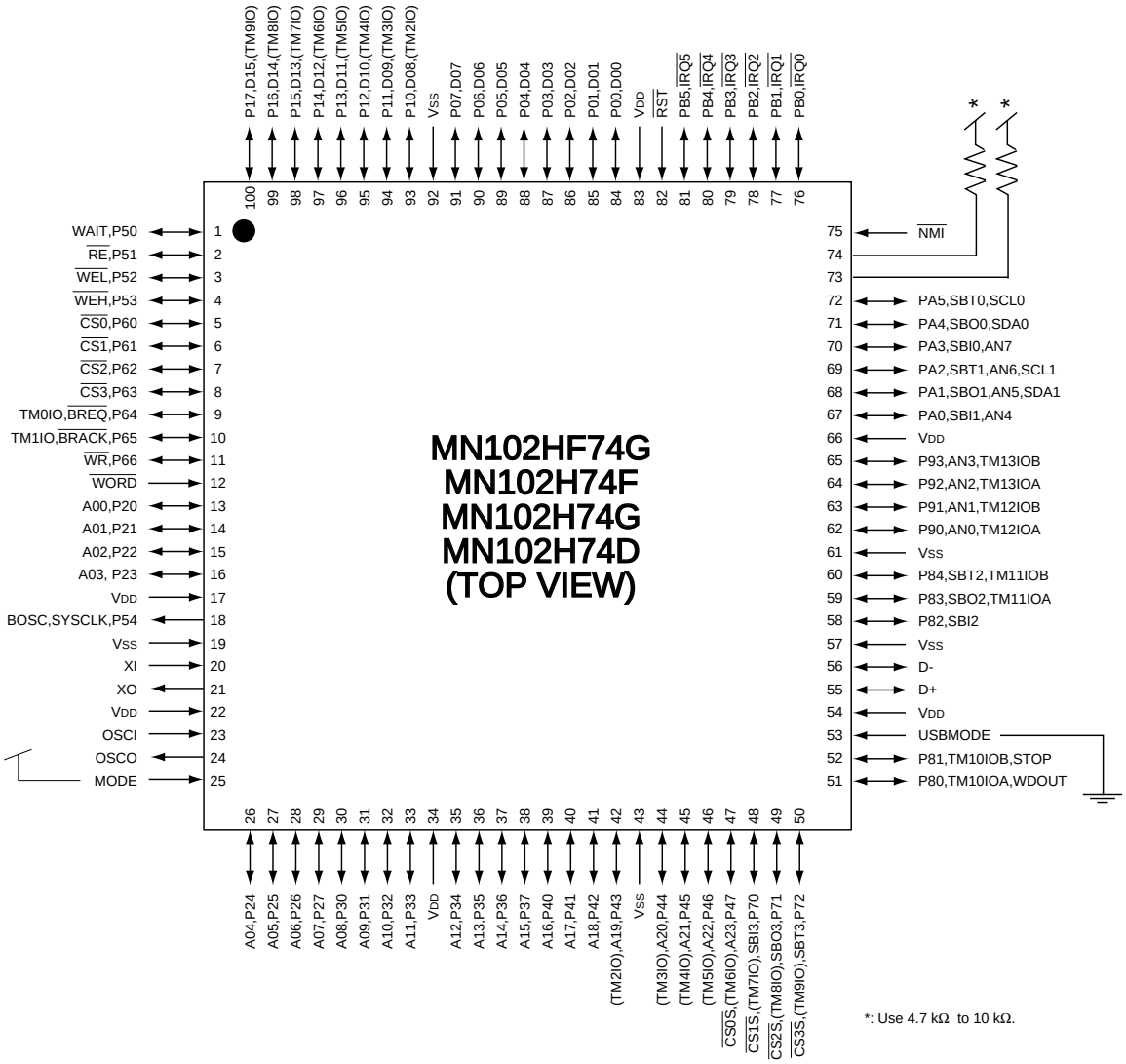


Figure 1-4-2 Pin Configuration in Memory Expansion Mode

> Unused pins require handling in the circuit (input pins are connected to VDD/VSS, output pins are open and input/output pins are connected to VDD/VSS or leave open depending on pin direction).

1-4-3 Processor Mode

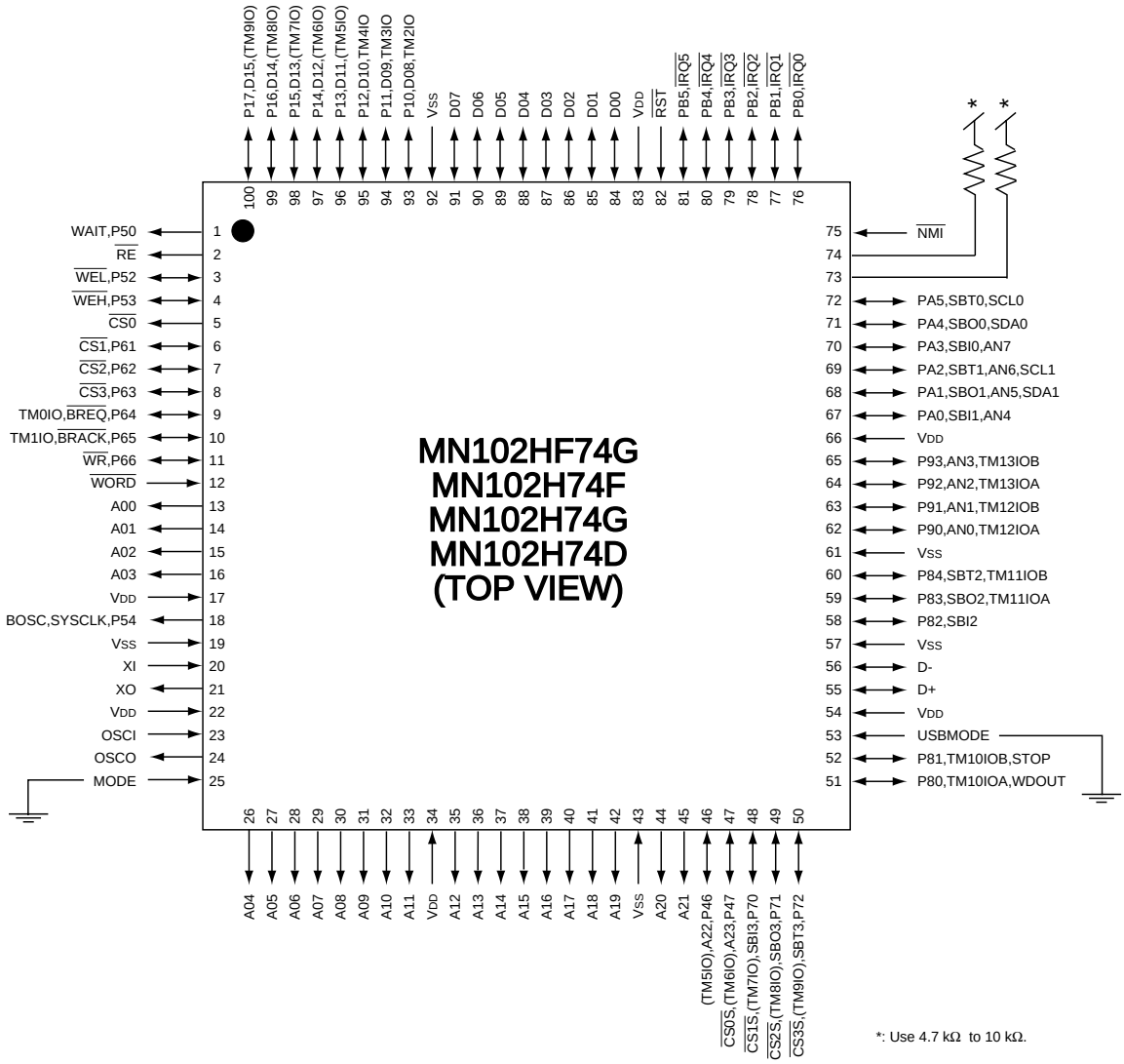


Figure 1-4-3 Pin Configuration in Processor Mode

Unused pins require handling in the circuit (input pins are connected to VDD/VSS, output pins are open and input/output pins are connected to VDD/VSS or leave open depending on pin direction).

1-4-4 List of Pin Functions

Table 1-4-1 List of Pin Functions (1/12)

Pin Number	Pin Name	Input/Output	Function	Description
17 22 34 54 66 83	V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD}		Power Power Power Power Power Power	There are six VDD pins. These pins must be connected to the power supply of 3.0 V to 3.6 V.
19 43 57 61 92	V_{SS} V_{SS} V_{SS} V_{SS} V_{SS}		Power (Ground) Power (Ground) Power (Ground) Power (Ground) Power (Ground)	There are five V_{SS} pins. These pins must be connected to the power supply of 0 V.
23 24	OSCI OSCO	Input Output	High-speed Oscillation Input High-speed Oscillation Output	For a self-excited oscillator configuration, connect these pins to crystal or ceramic oscillators. A feedback resistor is built in between these two pins. For stability, insert a capacitor of 20 pF to 33 pF between OSCI pin or OSCO pin and V_{SS} pin. (For the exact capacitance, consult the oscillator manufacturer). For an external oscillator configuration, input a pulse of 12 MHz with the amplitude of V_{DD} and V_{SS} to the OSCI pin. Leave the OSCO pin open. “Refer to Figure 1-4-4” Connecting OSCO pin to the external circuit directly is not allowed when the oscillation clock is taken from the MN102H74G/74F/74D/F74G. Select the BOSC pin for a synchronous signal.
20 21	XI XO	Input Output	Low-speed Oscillation Input Low-speed Oscillation Output	For a self-excited oscillator configuration, connect these pins to crystal or ceramic oscillators. A feedback resistor is built in between these two pins. For stability, insert a capacitor of 100 pF to 200 pF between XI pin or XO pin and V_{SS} pin. (For the exact capacitance, consult the oscillator manufacturer). For an external oscillator configuration, input a pulse of 32 kHz to 166 kHz with the amplitude of V_{DD} and V_{SS} to the XI pin. Leave the XO pin open. Refer to “Figure 1-4-5”. When these pins are unused, connect the XI pin to V_{DD} pin or V_{SS} pin and leave XO pin open. Connecting XO pin to the external circuit directly is not allowed when the oscillation clock is taken from the MN102H74G/74F/74D/F74G. Select the BOSC pin for a synchronous signal.
82	$\overline{RST}$	Input	Reset Input	$\overline{RST}$ pin resets the MN102H74G/74F/74D/F74G. With a 12-MHz oscillator, reset starts when the low level is input to this pin for more than 167 ns. Reset starts even when the noise is input to this pin for 167 ns or less. When the high level is input to this pin, reset is released. After this pin is high level, the oscillation waits of the high-speed oscillation pins (OSCI and OSCO) are performed (for approximately 5.481 ms with a 12-MHz oscillator). After than, the MN102H74G/74F/74D/F74G starts executing the instruction on x’080000’. Refer to “Figure 1-4-6”.

Table 1-4-1 List of Pin Functions (2/12)

Pin Number	Pin Name	Input/Output	Function	Description
18	BOSC	Output	System Clock 1 Output	This pin provides the system clock. After reset release, the pin always outputs BOSC. When the high-speed oscillation pin is operating at 12 MHz, the pin outputs the clock of 24 MHz.
	SYSCLK	Output	System Clock 2 Output	This pin also provide another system clock (SYSCLK). This pin can outputs SYSCLK by setting P5MD register.
	P54	Output	General-purpose Port 54 Output	When this pin is not used as BOSC or SYSCLK, it can be used as general-purpose 53 output port. Setting P5MD register switches the function. Refer to “Chapter 11 Ports”.
25	MODE	Input	Mode Setup Input	This pin selects the processor mode or single-chip mode (Memory expansion mode). Pulling this pin low selects processor mode. The internal ROM space is connected to external memory space and the CPU starts executing the instruction on x’080000’ of the external memory. Pulling this pin high selects the single-chip mode (memory expansion mode). The CPU starts executing the instruction on x’080000’ of internal ROM space. In the memory expansion mode, set the port mode register to address output, data output and , memory control signal output with the instruction. Do not change the mode by setting this pin during operation. Operation cannot be guaranteed if the mode is changed by this pin during operation. Refer to “Chapter 3 Bus Interface”.
12	$\overline{\text{WORD}}$	Input	Data Bus Width Setup Input	This pin selects 8-bit data bus width or 16-bit data bus width for external memory space 0 immediately after reset release in the processor mode or the memory expansion mode. Pulling this pin high selects 8-bit data bus width while pulling this pin low selects 16-bit data bus width. Use this pin as a data bus width setup pin in the processor mode or the memory expansion mode. The MEMMDn(S) register sets the data bus width for external memory space 1 to external memory space 3. After reset release, the MEMMDn(S) register sets the data bus width for external memory space 1 to external memory space 3 regardless of this pin level. Refer to “Chapter 3 Bus Interface”.
	P67	Input/Output	General-purpose Port 67	In the single-chip mode, this pin can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pull-up resistor controlled by software. Refer to “Chapter 11 Ports”.
1	WAIT	Input	Bus Cycle Wait Input	This pin extends the access cycle to the external memory when the waits for the external memory is set to handshake mode in the processor mode or the memory expansion mode. Pulling this pin low ends the access to the external memory. Refer to “Chapter 3 Bus Interface”.
	P50	Input/Output	General-purpose Port 50	When this pin is not used as a WAIT pin, this pin can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pull-up resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (3/12)

Pin Number	Pin Name	Input/Output	Function	Description
2	$\overline{RE}$	Output	Read Enable Output	This pin provides a control signal for the external memory read in processor mode or memory expansion mode. When connecting to SRAM or ROM, connect the $\overline{RE}$ pin to the $\overline{OE}$ pin on the external memory. The $\overline{RE}$ pin outputs low level during read operation and the memory contents to the CPU. Refer to “Chapter 3 Bus Interface”. This pin will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P51	Input/Output	General-purpose Port 51	When this pin is not used as the $\overline{RE}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
3	$\overline{WEL}$	Output	Lower Byte Write Enable Output	This pin provides a control signal for the external memory write in processor mode or memory expansion mode. When connecting to SRAM, connect the $\overline{WEL}$ pin to the $\overline{WE}$ pin on the external memory. The $\overline{WEL}$ pin outputs low level while it writes the data to the lower bytes (bit 7 to bit 0) and then loads the data to the external memory Refer to “Chapter 3 Bus Interface”. This pin will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P52	Input/Output	General-purpose Port 52	When this pin is not used as the $\overline{WEL}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
4	$\overline{WEH}$	Output	Upper Byte Write Enable Output	This pin provides a control signal for the external memory write in processor mode or memory expansion mode. When connecting to SRAM, connect the $\overline{WEH}$ pin to the $\overline{WE}$ pin on the external memory. The $\overline{WEH}$ pin outputs low level while it writes the data to the upper bytes (bit 15 to bit 8) and then loads the data to the external memory Refer to “Chapter 3 Bus Interface”. This pin will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P53	Input/Output	General-purpose Port 53	When this pin is not used as the $\overline{WEH}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (4/12)

Pin Number	Pin Name	Input/Output	Function	Description
5	$\overline{CS0}$	Output	Chip Select Output	This pin provides a chip select signal corresponding to each external memory space when accessing SRAM or ROM connected to the external memory space in processor mode or memory expansion mode. Connect the CS0 ($\overline{CS0S}$) pin to the $\overline{CS}$ pin on the external memory. This pin does not provide a chip select signal (CS0, $\overline{CS0S}$) when accessing internal ROM, internal RAM ot internal I/O register space. Refer to “Chapter 3 Bus Interface”. This pin will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P60	Input/Output	General-purpose Port 60	When this pin is not used as the $\overline{CS0}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
6	$\overline{CS1}$	Output	Chip Select Output	Refer to the above description of pin 5 $\overline{CS0}$ pin for details.
	P61	Input/Output	General-purpose Port 61	When this pin is not used as the $\overline{CS1}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
7	$\overline{CS2}$	Output	Chip Select Output	Refer to the above description of pin 5 $\overline{CS0}$ pin for details.
	P62	Input/Output	General-purpose Port 62	When this pin is not used as the $\overline{CS2}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
8	$\overline{CS3}$	Output	Chip Select Output	Refer to the above description of pin 5 $\overline{CS0}$ pin for details.
	P63	Input/Output	General-purpose Port 63	When this pin is not used as the $\overline{CS3}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
9	$\overline{BREQ}$	Input	Bus Request Input	$\overline{BREQ}$ pin and $\overline{BRACK}$ pin operate bus arbitration. Pulling the $\overline{BREQ}$ pin low suspends the current instruction execution, makes address, data and control signals in a high impedance state and opens the bus. After that, the $\overline{BRACK}$ pin outputs low level. The $\overline{BREQ}$ pin, however, opens the bus after the bus access is completed if the MN102H74G/74F/74D/F74G accesses the bus. Pulling the $\overline{BREQ}$ pin high returns the bus.
	TM0IO	Input/Output	Timer 0 Input/Output	This pin is also used as a timer 0 input/output pin. Refer to “Chapter 5 Timers”.
	P64	Input/Output	General-purpose Port 64	When this pin is not used as $\overline{BREQ}$ or TM0IO, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (5/12)

Pin Number	Pin Name	Input/Output	Function	Description
10	$\overline{\text{BRACK}}$	Output	Bus Request Enable	$\overline{\text{BREQ}}$ pin and $\overline{\text{BRACK}}$ pin operate bus arbitration. Refer to the description of pin 9 $\overline{\text{BREQ}}$ for details.
	TM1IO	Input/Output	Timer 1 Input/Output	This pin is also used as a timer 1 input/output pin. Refer to “Chapter 5 Timers”.
	P65	Input/Output	General-purpose Port 65	When this pin is not used as $\overline{\text{BRACK}}$ or TM1IO, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
11	$\overline{\text{WR}}$	Output	External Access Direction Output	This pin provides an access direction signal for the external memory in processor mode or memory expansion mode. When connecting SRAM used byte data control mode, connect the $\overline{\text{WR}}$ pin to the $\overline{\text{WE}}$ pin on the external memory. The $\overline{\text{WR}}$ pin outputs low level during write operation and loads the data to the external memory. Refer to “Chapter 3 Bus Interface”. This pin will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P66	Input/Output	General-purpose Port 66	When this pin is not used as the $\overline{\text{WR}}$ pin, it can be used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
13 to 16 26 to 29	A0 to A7	Output	Address Output	These pins output the addresses of the external memory in processor mode or memory expansion mode. Connect these pins to the address pins on the external memory or address decode circuit. The values of the pins are undefined when these pins are not connected to the external memory. Refer to “Chapter 3 Bus Interface”. These pins will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P20 to P27	Input/Output	General-purpose Ports 20 to 27	When these pins are not used as address output pins, they are used as general-purpose input/output ports. The input/output direction is controlled in bit units. These pins have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.
30 to 33 35 to 38	A8 to A15	Output	Address Output	Refer to the above description of A0 to A7 pins for details.
	P30 to P37	Input/Output	General-purpose Ports 30 to 37	When these pins are not used as address output pins, they are used as general-purpose input/output ports. The input/output direction is controlled in bit units. These pins have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (6/12)

Pin Number	Pin Name	Input/Output	Function	Description
39 to 41	A16 to A18	Output	Address Output	Refer to the description of A0 to A7 pins for details.
	P40 to P42	Input/Output	General-purpose Ports 40 to 42	When these pins are not used as address output pins, they are used as general-purpose input/output ports. The input/output direction is controlled in bit units. These pins have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.
42, 44 to 46	A19 to A22	Output	Address Output	Refer to the description of A0 to A7 pins for details.
	TM2IO to TM5IO	Input/Output	Timer 2 to 5 Input/ Output	These pins can be used as timer 2 to 5 input/output pins. Pins 93 to 96 have the same functions. Refer to “Chapter 5 Timers”.
	P43 to P46	Input/Output	General-purpose Ports 43 to 46	When these pins are not used as address output pins or timer input/ output pins, they are used as general-purpose input/output ports. The input/output direction is controlled in bit units. These pins have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.
47	A23	Output	Address Output	Refer to the description of A0 to A7 pins for details.
	$\overline{\text{CS0S}}$	Output	Chip Select Output	Refer to the description of pin 5 $\overline{\text{CS0}}$ for details.
	TM6IO	Input/Output	Timer 6 Input/Output	This pin can be used as a timer 6 input/output pin. Pin 97 has the same function. Refer to “Chapter 5 Timers”.
	P47	Input/Output	General-purpose Port 47	When this pin is not used as address output pin, $\overline{\text{CS0S}}$ or TM6IO, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
48	$\overline{\text{CS1S}}$	Output	Chip Select Output	Refer to the description of pin 5 $\overline{\text{CS0}}$ for details.
	TM7IO	Input/Output	Timer 7 Input/Output	This pin can be used as a timer 7 input/output pin. Pin 98 has the same function. Refer to “Chapter 5 Timers”.
	SBI3	Input	Serial Interface 3 Data Input	This pin can be used as a data input pin of serial interface 3. Refer to “Chapter 6 Serial Interface”.
	P70	Input/Output	General-purpose Port 70	When this pin is not used as $\overline{\text{CS1S}}$, TM7IO or SBI3, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (7/12)

Pin Number	Pin Name	Input/Output	Function	Description
49	$\overline{CS2S}$	Output	Chip Select Output	Refer to the description of pin 5 $\overline{CS0}$ for details.
	TM8IO	Input/Output	Timer 8 Input/Output	This pin can be used as a timer 8 input/output pin. Pins 99 has the same function. Refer to “Chapter 5 Timers”.
	SBO3	Output	Serial Interface 3 Data Output	This pin can be used as a data output pin of serial interface 3. Refer to “Chapter 6 Serial Interface”.
	P71	Input/Output	General-purpose Port 71	When this pin is not used as $\overline{CS2S}$, TM8IO or SBO3, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
50	$\overline{CS3S}$	Output	Chip Select Output	Refer to the description of pin 5 $\overline{CS0}$ for details.
	TM9IO	Input/Output	Timer 9 Input/Output	This pin can be used as a timer 9 input/output pin. Pins 100 has the same function. Refer to “Chapter 5 Timers”.
	SBT3	Input/Output	Serial Interface 3 Clock Input/Output	This pin can be used as a synchronous transfer clock input/output pin of serial interface 3. Refer to “Chapter 6 Serial Interface”.
	P72	Input/Output	General-purpose Port 72	When this pin is not used as $\overline{CS3S}$, TM9IO or SBT3, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
51	WDOUT	Output	Watchdog timer Overflow Output	This pin outputs a high pulse when the watchdog timer overflows.
	TM10IOA	Input/Output	Timer 10A Input/Output	This pin can be used as a timer 10 input capture A input pin or a timer 10 output compare A output pin. Refer to “Chapter 5 Timers”.
	P80	Input/Output	General-purpose Port 80	When this pin is not used as WDOUT or TM10IOA, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (8/12)

Pin Number	Pin Name	Input/Output	Function	Description
52	STOP	Output	STOP State Output	This pin outputs a high pulse when the STOP mode is selected.
	TM10IOB	Input/Output	Timer 10B Input/Output	This pin can be used as a timer 10 input capture B input pin or a timer 10 output compare B output pin. Refer to “Chapter 5 Timers”.
	P81	Input/Output	General-purpose Port 81	When this pin is not used as STOP or TM10IOB, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
53	USBMODE	Input	USB Mode Setup Input on ICE	This pin selects the mode which inputs 48-MHz clock for USB function from external device by pulling this pin high in the ICE system. Normally, pull this pin low.
55	D+	Input/Output	USB D+	Connect the USB D+ pin to a 24 Ω resistor in series.
56	D-	Input/Output	USB D-	Connect the USB D- pin to a 24 Ω resistor in series.
58	SBI2	Input	Serial Interface 2 Data Input	This pin can be used as a data input pin of serial interface 2. Refer to “Chapter 6 Serial Interface”.
	P82	Input/Output	General-purpose Port 82	When this pin is not used as SBI2, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
59	TM11IOA	Input/Output	Timer 11A Input/Output	This pin can be used as a timer 11 input capture A input pin or a timer 11 output compare A output pin. Refer to “Chapter 5 Timers”.
	SBO2	Output	Serial Interface 2 Data Output	This pin can be used as a data output pin of serial interface 2. Refer to “Chapter 6 Serial Interface”.
	P83	Input/Output	General-purpose Port 83	When this pin is not used as TM11IOA or SBO2, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (9/12)

Pin Number	Pin Name	Input/Output	Function	Description
60	TM11IOB	Input/Output	Timer 11B Input/Output	This pin can be used as a timer 11 input capture B input pin or a timer 11 output compare B output pin. Refer to “Chapter 5 Timers”.
	SBT2	Input/Output	Serial Interface 2 Clock Input/Output	This pin can be used as a synchronous transfer clock input/output pin of serial interface 2. Refer to “Chapter 6 Serial Interface”.
	P84	Input/Output	General-purpose Port 84	When this pin is not used as TM11IOB or SBT2, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
62	TM12IOA	Input/Output	Timer 12A Input/Output	This pin can be used as a timer 12 input capture A input pin or a timer 12 output compare A output pin. Refer to “Chapter 5 Timers”.
	AN0	Input	A/D 0 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	P90	Input/Output	General-purpose Port 90	When this pin is not used as TM12IOA or AN0, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
63	TM12IOB	Input/Output	Timer 12B Input/Output	This pin can be used as a timer 12 input capture B input pin or a timer 12 output compare B output pin. Refer to “Chapter 5 Timers”.
	AN1	Output	A/D 1 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	P91	Input/Output	General-purpose Port 91	When this pin is not used as TM12IOB or AN1, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
64	TM13IOA	Input/Output	Timer 13A Input/Output	This pin can be used as a timer 13 input capture A input pin or a timer 13 output compare A output pin. Refer to “Chapter 5 Timers”.
	AN2	Output	A/D 2 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	P92	Input/Output	General-purpose Port 92	When this pin is not used as TM13IOA or AN2, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (10/12)

Pin Number	Pin Name	Input/Output	Function	Description
65	TM13IOB	Input/Output	Timer 13B Input/Output	This pin can be used as a timer 13 input capture B input pin or a timer 13 output compare B output pin. Refer to “Chapter 5 Timers”.
	AN3	Input/Output	A/D 3 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	P93	Input/Output	General-purpose Port 93	When this pin is not used as TM13IOB or AN3, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
67	SBI1	Input	Serial Interface 1 Data Input	This pin can be used as a data input pin of serial interface 1. Refer to “Chapter 6 Serial Interface”.
	AN4	Input	A/D 4 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	PA0	Input/Output	General-purpose Port A0	When this pin is not used as SBI1 or AN4, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
68	SBO1	Output	Serial Interface 1 Data Output	This pin can be used as a data output pin of serial interface 1. Refer to “Chapter 6 Serial Interface”.
	SDA1	Input/Output	Serial Interface 1 Data Input/Output	This pin can be used as a I^2C data input/output pin of serial interface 1. Refer to “Chapter 6 Serial Interface”.
	AN5	Input	A/D 5 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	PA1	Input/Output	General-purpose Port A1	When this pin is not used as SBO1,SDA1 or AN5, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.
69	SBT1	Input/Output	Serial Interface 1 Clock Input/Output	This pin can be used as a synchronous transfer clock input/output pin of serial interface 1. Refer to “Chapter 6 Serial Interface”.
	SCL1	Output	Serial Interface 1 Clock Output	This pin can be used as a I^2C clock signal output pin of serial interface 1. Refer to “Chapter 6 Serial Interface”.
	AN6	Input	A/D 6 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to “Chapter 7 Analog Interface”.
	PA2	Input/Output	General-purpose Port A2	When this pin is not used as SBT1,SCL1 or AN6, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to “Chapter 11 Ports”.

Table 1-4-1 List of Pin Functions (11/12)

Pin Number	Pin Name	Input/Output	Function	Description
70	SBI0	Input/Output	Serial Interface 0 Data Input	This pin can be used as a data input pin of serial interface 0. Refer to "Chapter 6 Serial Interface".
	AN7	Input/Output	A/D 7 Conversion Input	This pin can be used as an A/D conversion input pin. Refer to "Chapter 7 Analog Interface".
	PA3	Input/Output	General-purpose Port A3	When this pin is not used as SBI0 or AN7, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to "Chapter 11 Ports".
71	SBO1	Output	Serial Interface 0 Data Output	This pin can be used as a data output pin of serial interface 0. Refer to "Chapter 6 Serial Interface".
	SDA0	Input/Output	Serial Interface 0 Data Input/Output	This pin can be used as a I^2C data input/output pin of serial interface 0. Refer to "Chapter 6 Serial Interface".
	PA4	Input/Output	General-purpose Port A4	When this pin is not used as SBO0 or SDA0, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to "Chapter 11 Ports".
72	SBT0	Input/Output	Serial Interface 0 Clock Input/Output	This pin can be used as a synchronous transfer clock input/output pin of serial interface 0. Refer to "Chapter 6 Serial Interface".
	SCL0	Output	Serial Interface 0 Clock Output	This pin can be used as a I^2C clock signal output pin of serial interface 0. Refer to "Chapter 6 Serial Interface".
	PA5	Input/Output	General-purpose Port A5	When this pin is not used as SBT0 or SCL0, it is used as a general-purpose input/output port. The input/output direction is controlled in bit units. This pin has a built-in pullup resistor controlled by software. Refer to "Chapter 11 Ports".
73 to 74	Pullup	Input	Pullup	These pins must be pulled up with 4.7 kΩ to 10 kΩ.
75	$\overline{\text{NMI}}$	Input	Nonmaskable Interrupt Input	This pin can be used as a $\overline{\text{NMI}}$ interrupt pin. A $\overline{\text{NMI}}$ interrupt occurs on the falling edge of low level. This pin also reads the general-purpose port B6 pin state.
76 to 81	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$	Input	External Interrupt Data Input	These pins can be used as external interrupt request input pins. Refer to "Chapter 4 Interrupts".
	PB0 to PB5	Input/Output	General-purpose Ports B0 to B5	These pins can be used as general-purpose input/output ports. The input/output direction is controlled in bit units. They have built-in pullup resistors controlled by software. Refer to "Chapter 11 Ports".

Table 1-4-1 List of Pin Functions (12/12)

Pin Number	Pin Name	Input/Output	Function	Description
84 to 91	D0 to D7	Input/Output	Data Input/Output	These pins input or output the lower 8-bit data of the external memory in processor mode or memory expansion mode. When the MN102H74G/74F/74D/F74G does not access the external memory, the direction of these pins become input. Refer to “Chapter 3 Bus Interface”. These pins will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	P00 to P07	Input/Output	General-purpose Ports 00 to 07	These pins can be used as general-purpose input/output ports in the single-chip mode. The input/output direction is controlled in bit units. They have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.
93 to 100	D8 to D15	Input/Output	Data Input/Output	These pins input or output the upper 8-bit data of the external memory in processor mode or memory expansion mode. When the MN102H74G/74F/74D/F74G does not access the external memory, the direction of these pins become input. Refer to “Chapter 3 Bus Interface”. These pins will be in a high impedance state during bus request, STOP mode or HALT mode. Refer to “12-3 List of Pin Functions”.
	TM2IO to TM9IO	Input/Output	Timer 2 to 9 Input/Output	These pins can be used as timer 2 input/output pin to timer 9 input/output pin. Pins 42, 44 to 50 have the same function. Refer to “Chapter 5 Timers”.
	P10 to P17	Input/Output	General-purpose Ports 10 to 17	When these pins are not used as D8 to D15 or TM2IO to TM9IO, they can be used as general-purpose input/output ports. The input/output direction is controlled in bit units. They have built-in pullup resistors controlled by software. Refer to “Chapter 11 Ports”.

■ Connection Examples of Power Pins, Oscillation Pins and Reset Pin

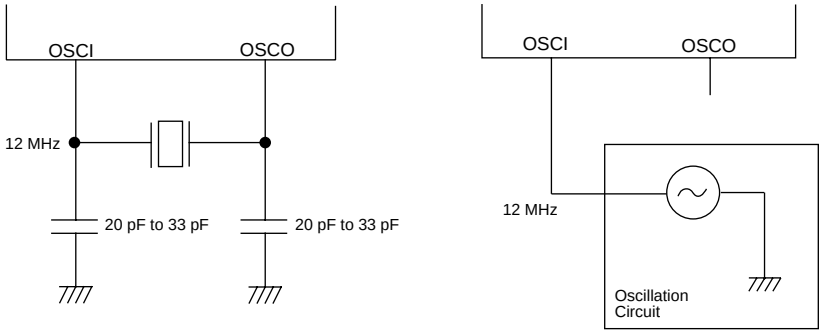


Figure 1-4-4 OSCI and OSCO Connection Example

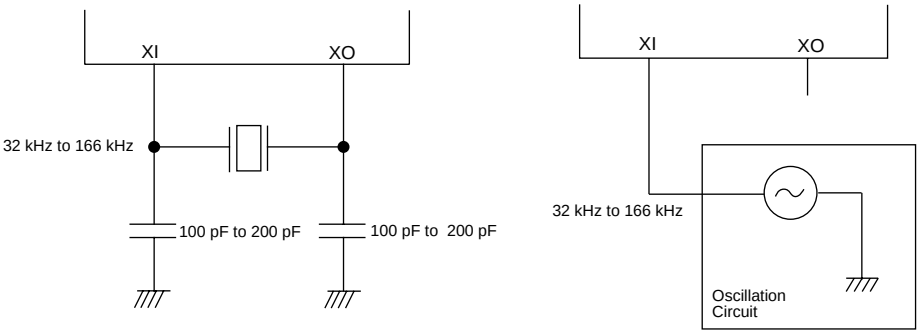


Figure 1-4-5 XI and XO Connection Example

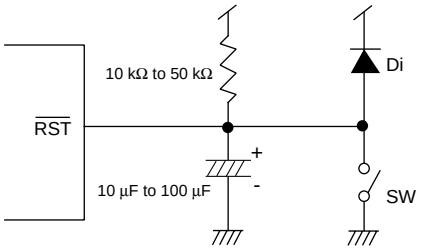
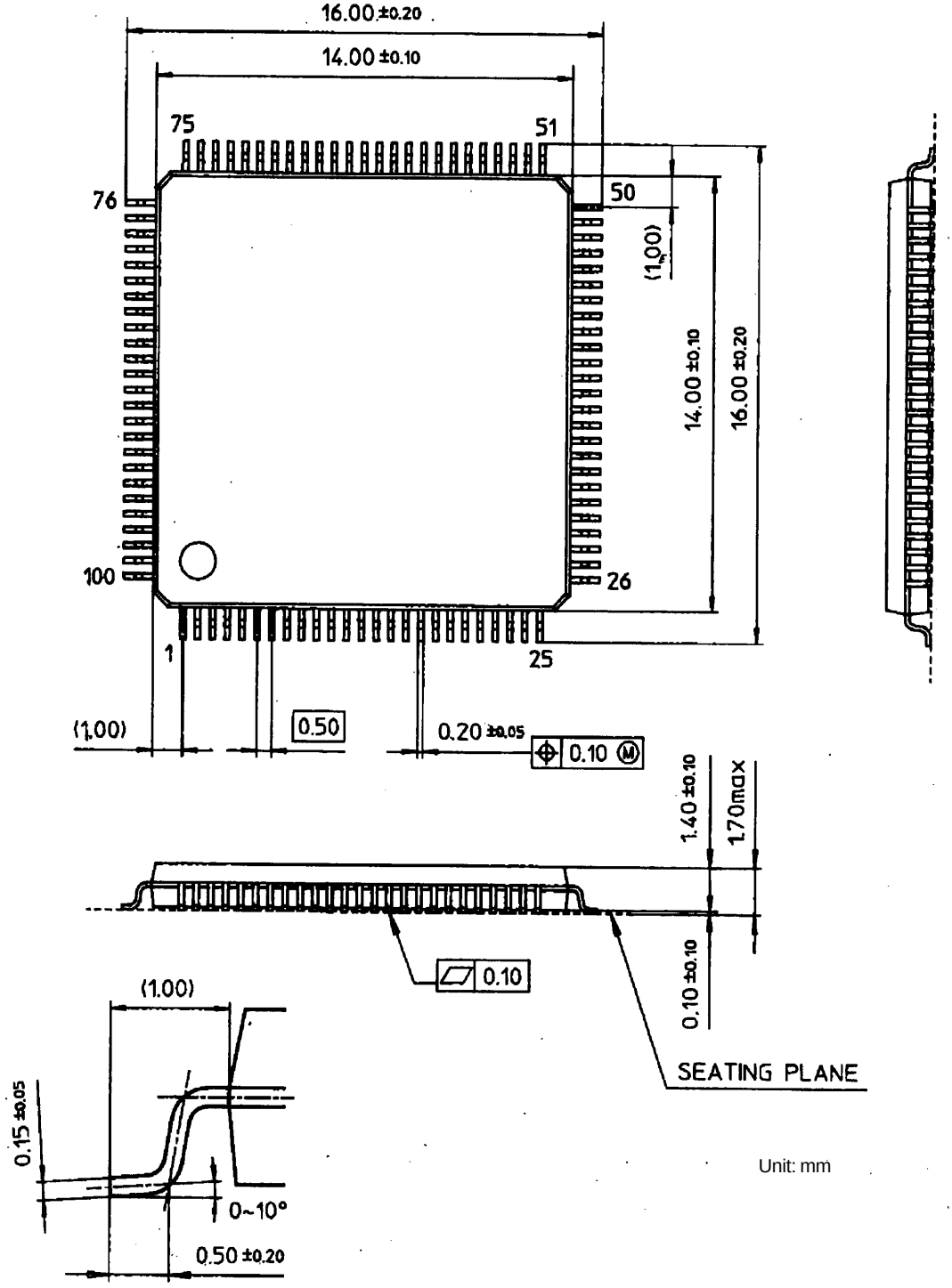


Figure 1-4-6 Reset Pin Connection Example

Package Code: LQFP100-P-1414



Body Material: Epoxy Resin Lead Material: Cu Alloy Lead Finish Method: Pd Plating

Figure 1-4-7 External Dimensions: 100-pin LQFP

> External dimensions are subject to change. Before using, please contact your nearest sales office for the latest product specifications.

1-5 Addressing Mode

The MN102H74G/74F/74D/F74G supports the following six addressing modes. All addressing modes are used with the data transfer instructions. Refer to “MN102H Series Instruction Manual”.

Table 1-5-1 Addressing Mode

Addressing Mode		Execution Address	Description
Register Direct	Dn An	——	Directly specifies internal registers.
Immidiate Value	imm8 imm16 imm24	——	Directly operates on the operand or mask value appended to the instruction code.
Register Indirect	(An)Note 1	23 0 An	Specificies the address using an address register.
Register Relative Indirect	(d8, An)Note 1 (d8, PC)Note 2	23 0 An/PC + d8 (signed extension)	Specificies the address using an address register or program counter with 8-bit displacement.
	(d16, An)Note 1 (d16, PC)Note 2	23 0 An/PC + d16 (signed extension)	Specificies the address using an address register or program counter with 16-bit displacement.
	(d24, An)Note 1 (d24, PC)Note 2	23 0 An/PC + d24	Specificies the address using an address register or program counter with 24-bit displacement.
Absolute	(abs16)Note 1	23 0 abs16(zero extension)	Specificies the address using the 16-bit value in the operand added to the instruction code.
	(abs24)Note 1	23 0 abs24	Specificies the address using the 24-bit value in the operand added to the instruction code.
Indexed Register Indirect	(Dn, An)Note 1	23 0 An + Dn	Specificies the address using an address register and data register.

Note1: The WORD access from the odd address is disable. Be careful of the effective address value on the MOV and MOVX instruction.
Note2: Branch instruction only

1-6 List of Instructions

The MN102H74G/74F/74D/F74G supports 41 assembler instructions. The byte data contains 8 bits, the word data contains 16 bits and the pointer type data contains 24 bits. Refer to “MN102H Series Instruction Manual”.

Table 1-6-1 List of Instructions (1/2)

Classification	Instruction	Operation
Data Transfer Instruction	MOV	Pointer type data transfer between registers Word data trasfer between data register and memory (with signed extension) Pointer type data trasfer between address register and memory Immediate value transfer to register
	MOVX	Pointer type data transfer between data register and memory
	MOVB	Byte data transfer to memory (with signed extension)
	MOVBU	Byte data transfer to memory (with zero extension)
	EXT	32-bit signed extension from word transfer
	EXTX	Signed extension from word data to pointer type data
	EXTXU	Zero extension from word data to pointer type data
	EXTXB	Signed extension from byte data to pointer type data
	EXTXBU	Zero extension from byte data to pointer type data
Arithmetic Operation Instruction	ADD	Addition
	ADDC	Addition with carry
	ADDNF	Addition without flag change
	SUB	Subtraction
	SUBC	Subtraction with borrow
	MUL	Signed multiplication
	MULU	Unsigned multiplication
	MULQ	High-speed signed multiplication (16 bits ×16 bits = 32 bits)
	MULQL	High-speed signed multiplication (16 bits ×116 bits = 24 bits)
	MULQH	High-speed signed multiplication (16 bits × 16 bits = 32 bits only the upper 16 bits of 32 bits are valid.)
	DIVU	Unsigned division
	CMP	Compare
Logical Operation Instruction	AND	Logical AND
	OR	Logical OR
	XOR	Exclusive logical OR
	NOT	Not (1's complement)
	ASR	Arithmetic right shift
	LSR	Logical right shift
	ROR	Right rotate
	ROL	Left rotate

Table 1-6-1 List of Instructions (2/2)

Classification	Instruction	Operation
Bit Manipulation Instructions	BTST	Bit test
	BSET	Bit test and set (process unit is byte)
	BCLR	Bit test and clear (process unit is byte)
Bit Test and Branch Instructions	TBcc	Bit test and conditional branch (only Z, NZ)
Branch Instructions	Bcc	Conditional branch (PC relative)
	Bccx	Conditional branch by expansion flag (PC relative)
	JMP	Unconditional branch (PC relative, register relative)
	JSR	Branch to subroutine (PC relative, register relative)
	RTS	Return from subroutine
	RTI	Return from interrupt
	NOP	No operation
Control Instructions	PXST	Saturated operation flag control instruction

Chapter 2 CPU Basics

2-1 Machine Clock

2-1-1 Machine Clock

The MN102H74G/74F/74D/F74G generates machine clock based on the system clock divided the base clock (OSCI/XI) by 2. The clock generator supplies the base clock. The CPU controls and executes based on this machine clock timing.

Since the external high-speed oscillation pin input is fixed to 12 MHz, OSCI is fixed to 24 MHz.

■ External Memory Access (No Wait)

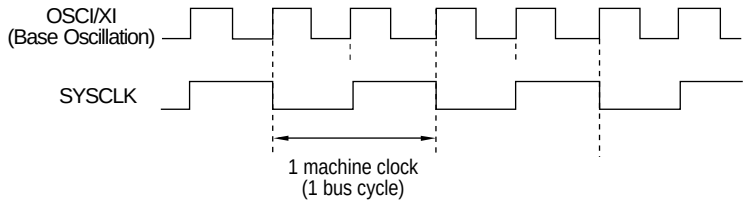


Figure 2-1-1 Machine Clock (No Wait)

■ External Memory Access (Wait)

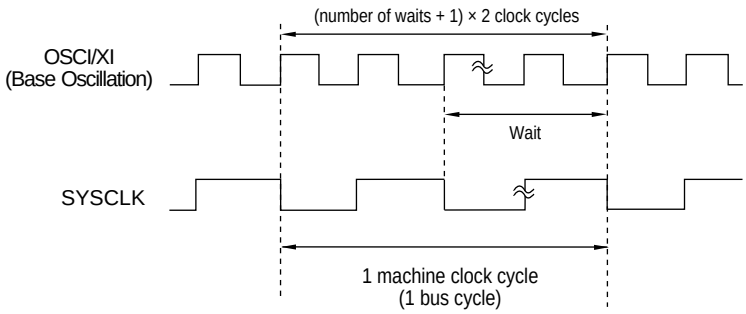


Figure 2-1-2 Machine Clock (Wait)

The 7-wait fixed mode is set at reset start.

The accesses of internal ROM and internal RAM connected to ROM bus and RAM bus are no waits. The access to internal I/O is 1 fixed wait. Extending ROM, RAM and I/O externally requires to insert the memory wait by setting the memory control register (MEMMDn).

The relation of SYSCLK and bus cycles may invert when the half wait is enabled.

■ Half Wait Enable

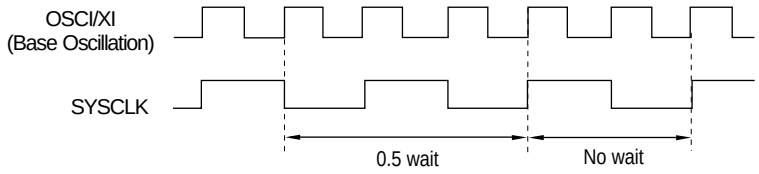


Figure 2-1-3 Clock Signal at Half Wait

The number of wait to be set requires extra 0.5 wait due to the synchronization with SYSCLK to access C-bus interface when the memory space in the half wait enable mode is selected.

■ C-Bus Access at Half Wait Enable

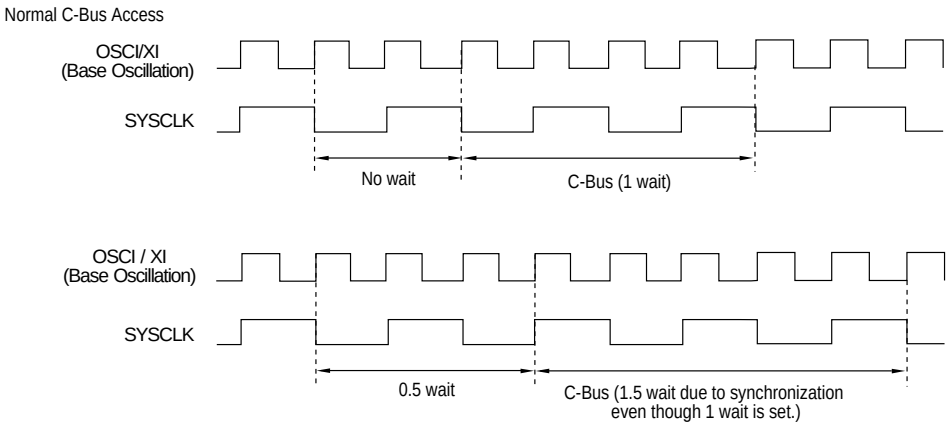


Figure 2-1-4 Clock Signal at Half Wait (C-Bus Access)

2-2 Instruction Execution Controller

2-2-1 Configuration

The instruction execution controller consists of four blocks including memory, instruction queue, instruction registers and instruction decoder.

Instructions are fetched in 2-byte units and temporarily stored in 4-byte instruction queue. The CPU transfers the data in 1-byte units from the instruction queue to the instruction register, and decodes the data in 1-byte units by the instruction decoder.

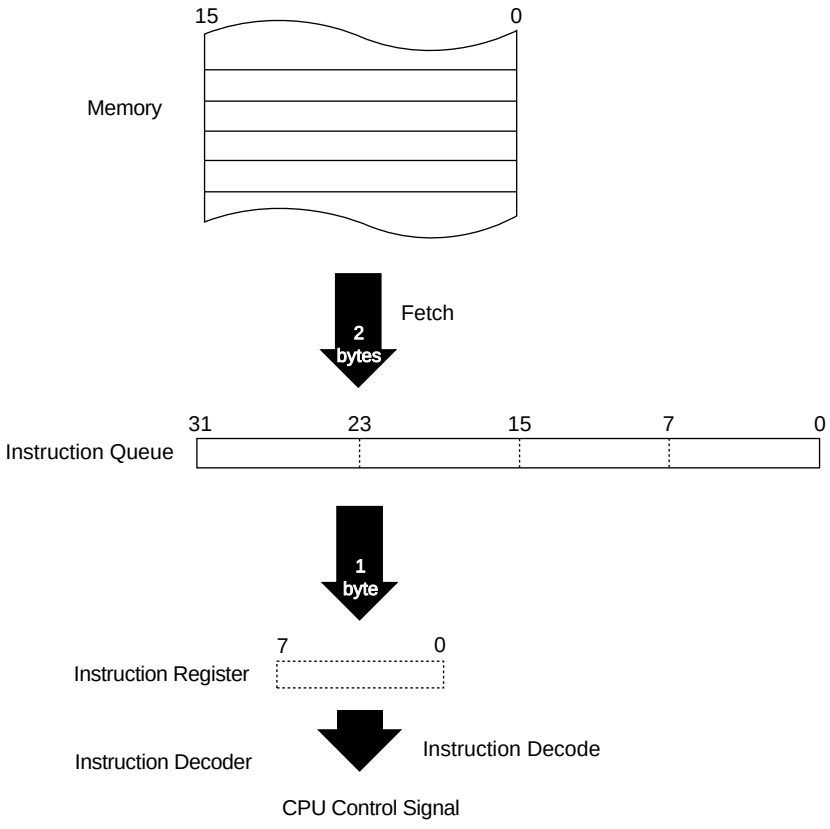


Figure 2-2-1 Instruction Execution Controller

2-2-2 Pipeline Process

The MN102H74G/74F/74D/F74G executes instructions in a 3-stage pipeline process of fetch, decode and execute. This parallel 3-stage pipeline process allows continuous instruction execution and improves instruction execution speed.

Instruction queue is a 4-byte buffer in advance of instructions. The instruction queue controls to fetch the next instruction when there is an empty between instruction queue and bus at each cycle on instruction execution.

The CPU stores the operation code of the next instruction to an instruction register at the last cycle of instruction. At this point, when all data needed for instruction queue is stored to instruction queue, the CPU can execute the next instruction immediately even if the direct address (abs) or the immediate value data (imm) is needed at the first cycle of the next instruction.

Operation code is the first word of the instruction to be executed.

Example 1

Address		
10	ADD	D0, D1
11	SUB	D1, D2
12, 13	ADD	imm8, D2
14	SUB	D0, D1
15, 16	ASR	D1

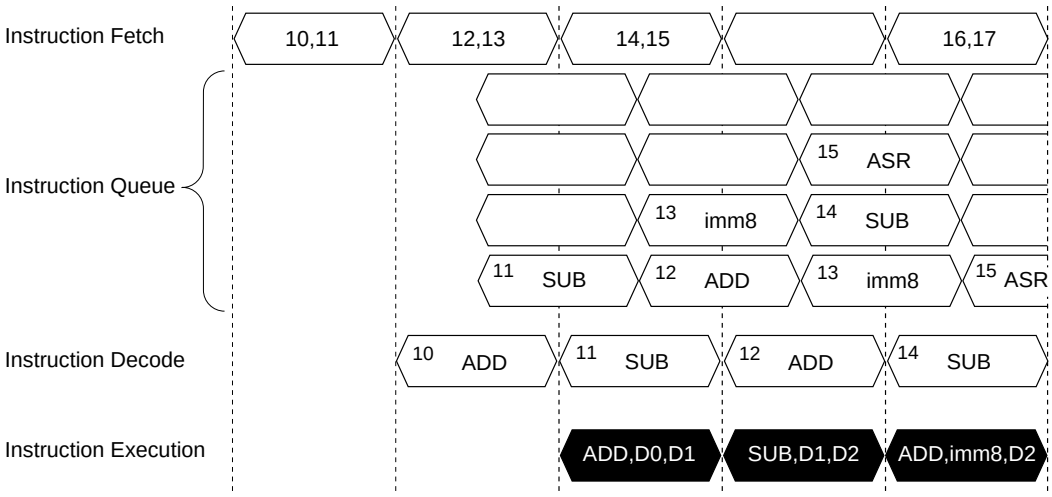


Figure 2-2-2 Pipeline Process Example 1

However, when the instruction queue is empty or all data such as abs and imm needed for instruction queue are not stored to the instruction queue, the instruction queue waits at least 1 cycle.

Example 2

```
          Address
          10, 11                    BRA   Label 1
          12                        MOV   D0, D1
           :
          100      Label 1          ADD    D1, D2
          101                         :
          102                         :
          103                         :
```

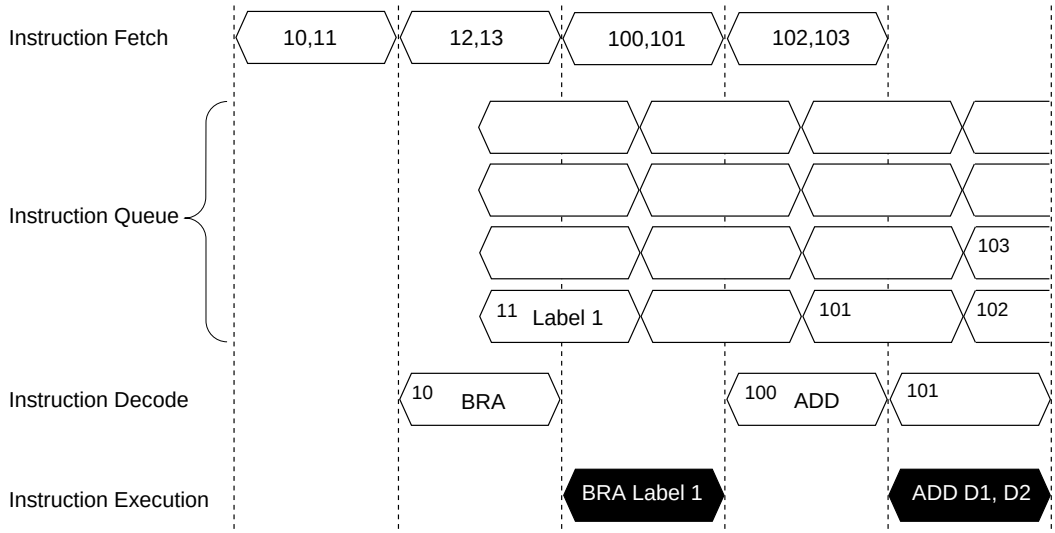


Figure 2-2-3 Pipeline Process Example 2

No consideration for programming is required because the instruction queue is controlled automatically by hardware. However, the operation of instruction queue should be considered to calculate the instruction execution time.

2-3 Internal Registers

2-3-1 Registers for Address

Registers for address consist of a program counter (PC) and address registers (An).

Registers are classified into internal registers in CPU core and special function registers. Internal registers include registers for address, registers for operation and a processor status word (PSW).

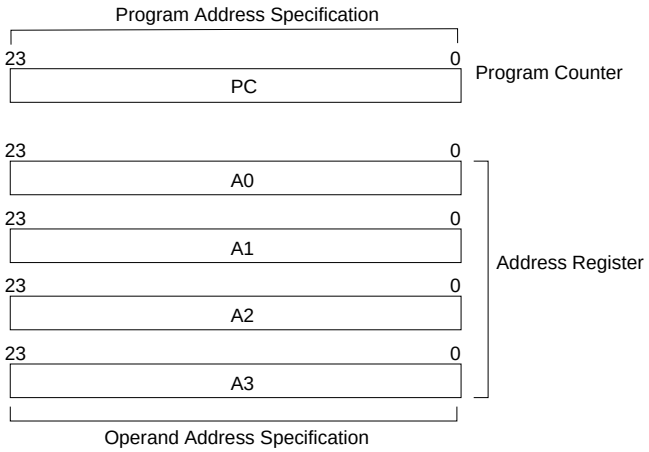


Figure 2-3-1 Address Register

The values of An registers are undefined at reset start. Set the initial values to the An registers by initialization program.

■ Program Counter (PC)

The program counter is a register to specify the memory address (24 bits) where the instruction in progress is stored. The program counter increments automatically each time the instruction is executed.

■ Address Registers (A0 - A3)

The address registers are 24-bit registers to specify data locations on memory. These registers support only addition/subtraction operations and comparison operations. Operations are executed in 24-bit units. The flag is changed in both 24 bits and lower 16 bits of address registers. The A3 register is assigned for stack pointer. Transfers between memory and these registers are always made in 24-bit units.

The A3 register is reserved for a stack pointer. Do not use the A3 register for other purposes.

2-3-2 Registers for Operation

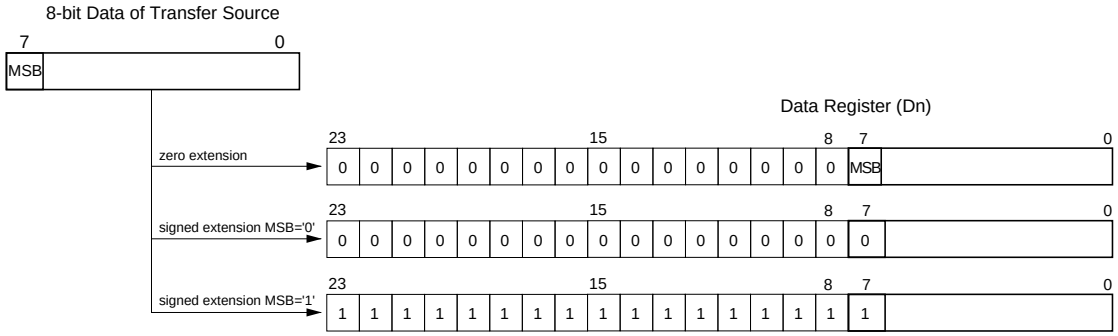
Registers for operation consist of a 16-bit multiplication/division register (MDR) and data registers (Dn).

Figure 2-3-2 Data Register

> The values of Dn registers and the MDR register are undefined at reset start. Set the initial values to the Dn registers the MDR register by initialization program.

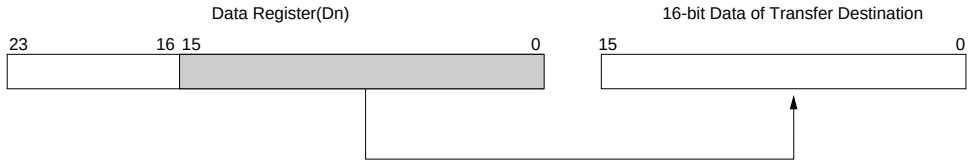
■ Data Registers (D0 - D3)

The data registers are 24-bit registers to support all arithmetic, logical and shift operations. All registers are used to transfer the data to memory. The byte-transfer from memory to registers executed with zero-extension (MOVBU instruction) or sign-extension (MOVB instruction). The lower 8 bits of the Dn register are transferred to memory.


8-bit Data of Transfer Source
7
0
MSB
Data Register (Dn)
zero extension
signed extension MSB='0'
signed extension MSB='1'
23
15
8
7
0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 MSB
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

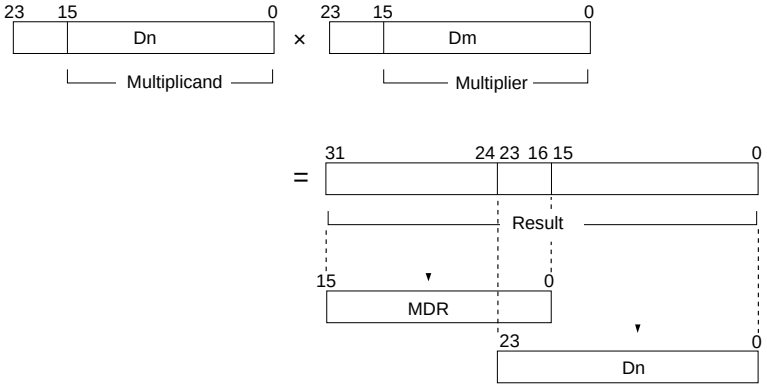
> The word-transfer from memory to data registers is executed with sign-extension (MOV instruction). The pointer data transfer with MOVX instruction is possible.

The lower 16 bits of the Dn register is transferred when the data is word-transferred to memory or 16-bit registers such as the MDR register.



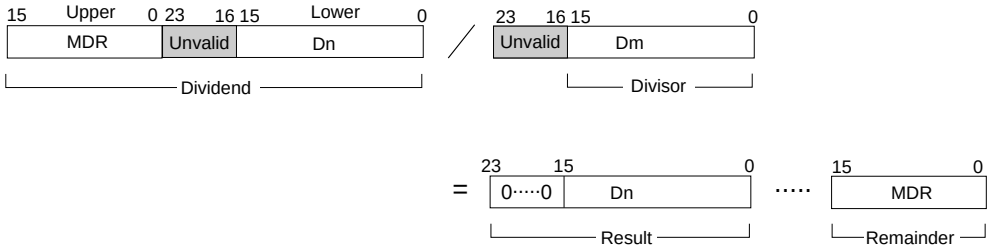
■ Multiplication/Division Register (MDR)

The multiplication/division register (MDR) is a 16-bit register reserved for multiplication and division operations. When the multiplication instruction is executed, the upper 16 bits of the 32-bit operation result are stored to the MDR register.



The same data is stored to the lower 8 bits of MDR and the upper 8 bits of Dn.

The 16-bit remainder of the division result is stored to the MDR register when the division instruction executed after the upper 16 bits of the 32-bit operation result are stored to the MDR register.



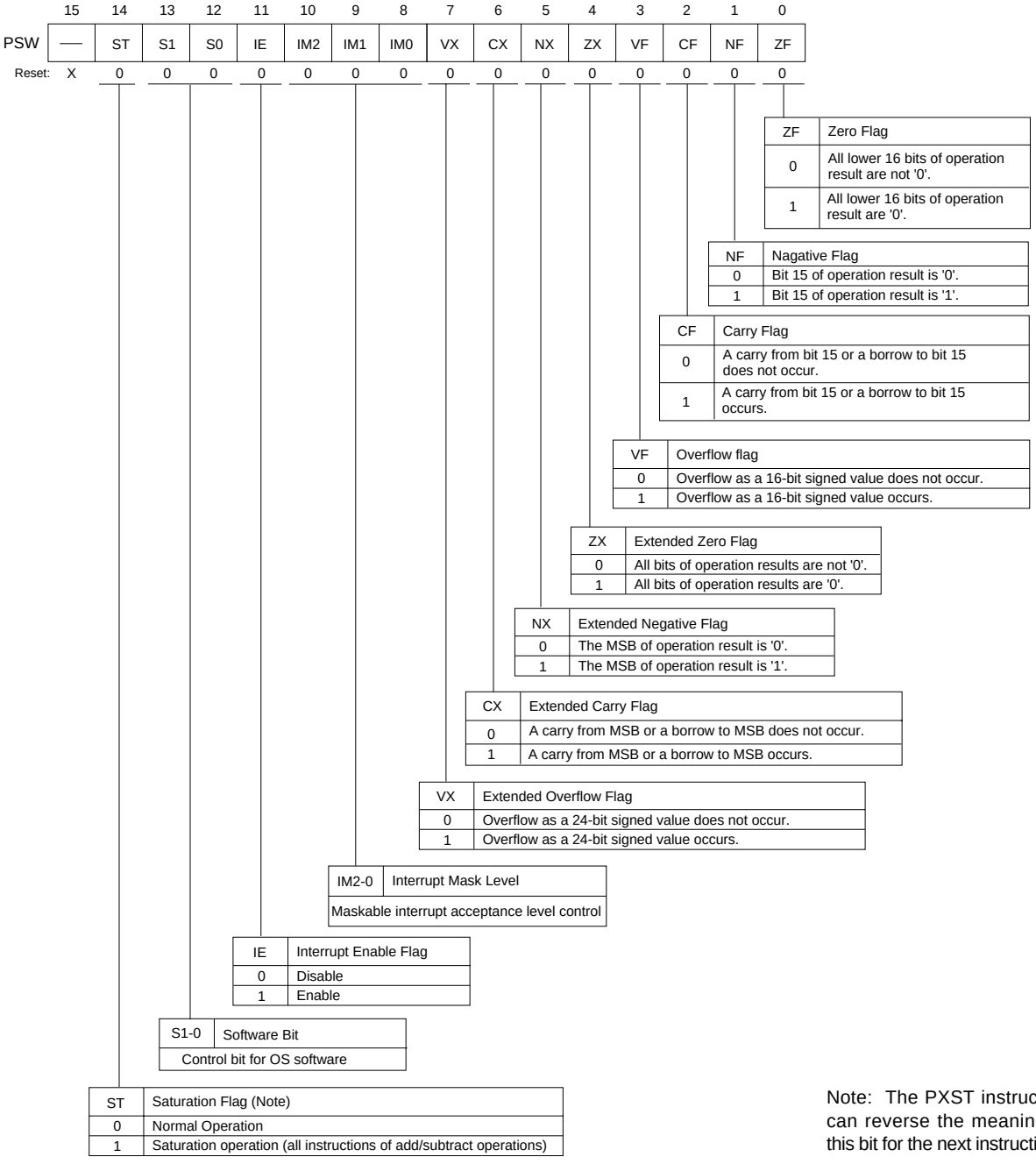
When the quotient cannot be expressed in 16 bits, the values of Dn and MDR become undefined and VX (overflow flag) = 1.

The quotient is zero-extended to 24 bits and is stored to Dn.

2-3-3 Processor Status Word

The processor status word (PSW) is a 16-bit register that stores the CPU interrupt status and flags for operation results.

The eight flags of ZF to VX indicate the operation results. The IMn flags and the IE flags indicate the interrupt control and become '0' after the CPU reset. The PSW is pushed on the stack area when an interrupt occurs and pulled when the CPU returns from the interrupt service routine.


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PSW — ST S1 S0 IE IM2 IM1 IM0 VX CX NX ZX VF CF NF ZF
Reset: X 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ZF Zero Flag
0 All lower 16 bits of operation result are not '0'.
1 All lower 16 bits of operation result are '0'.
NF Nagative Flag
0 Bit 15 of operation result is '0'.
1 Bit 15 of operation result is '1'.
CF Carry Flag
0 A carry from bit 15 or a borrow to bit 15 does not occur.
1 A carry from bit 15 or a borrow to bit 15 occurs.
VF Overflow flag
0 Overflow as a 16-bit signed value does not occur.
1 Overflow as a 16-bit signed value occurs.
ZX Extended Zero Flag
0 All bits of operation results are not '0'.
1 All bits of operation results are '0'.
NX Extended Negative Flag
0 The MSB of operation result is '0'.
1 The MSB of operation result is '1'.
CX Extended Carry Flag
0 A carry from MSB or a borrow to MSB does not occur.
1 A carry from MSB or a borrow to MSB occurs.
VX Extended Overflow Flag
0 Overflow as a 24-bit signed value does not occur.
1 Overflow as a 24-bit signed value occurs.
IM2-0 Interrupt Mask Level
Maskable interrupt acceptance level control
IE Interrupt Enable Flag
0 Disable
1 Enable
S1-0 Software Bit
Control bit for OS software
ST Saturation Flag (Note)
0 Normal Operation
1 Saturation operation (all instructions of add/subtract operations)

Note: The PXST instruction can reverse the meaning of this bit for the next instruction.

Figure 2-3-3 Processor Status Word

■ Zero Flag (ZF)

If all lower 16 bits are ‘0’ as the result of operation, ZF becomes ‘1’; otherwise ZF is ‘0’.

■ Negative Flag (NF)

If bit 15 becomes ‘1’ as the result of operation, NF becomes ‘1’; otherwise NF is ‘0’.

■ Carry Flag (CF)

If a carry from or a borrow to bit 15 occurs as the result of operation, CF becomes ‘1’; otherwise CF is ‘0’.

■ Overflow Flag (VF)

If an overflow in a 16-bit signed value occurs as the result of arithmetic operation, VF becomes ‘1’; otherwise VF is ‘0’.

■ Extended Zero Flag (ZX)

If all bits become ‘0’ as the result of operation, ZX becomes ‘1’; otherwise ZX is ‘0’.

■ Extended Negative Flag (NX)

If MSB become ‘1’ as the result of operation, NX becomes ‘1’; otherwise NX is ‘0’.

■ Extended Carry Flag (CX)

If a carry from or a borrow to MSB occurs as the result of operation, CX becomes ‘1’; otherwise CX is ‘0’.

■ Extended Overflow Flag (VX)

If an overflow in a 24-bit signed value occurs as the result of arithmetic operation, VX becomes ‘1’; otherwise VX is ‘0’.

PSW is executed with the MOV, OR, AND instructions. The flag meanings may be different by the instructions. Refer to the instruction manual for details.

■ Interrupt Mask Level (IM2 to IM0)

The interrupt mask level flags (IM2 to IM0) controls the maskable interrupt acceptance based on the interrupt vector priority. The interrupt mask level flags select from interrupt disabled (‘000’) to level 6 (‘111’) which level 0 is the highest priority mask level. When the interrupt level set by the interrupt level flags (ILVn) of the interrupt control register is higher than the interrupt level set by the interrupt mask level flags (IMn), the interrupt with the level set by ILVn flags is accepted. Once the interrupt is accepted, the level is set to the IMn flags.

Table 2-3-1 Interrupt Mask Level and Interrupt Acceptance

Interrupt Mask Level			Acceptable Interrupt Level	Priority
IM2	IM1	IM0		
0	0	0	Nonmaskable Interrupt (NMI) only	High
0	0	1	Level 0, NMI	
0	1	0	Level 0 to Level1, NMI	
0	1	1	Level 0 to Level 2, NMI	
1	0	0	Level 0 to Level 3, NMI	
1	0	1	Level 0 to Level 4, NMI	
1	1	0	Level 0 to Level 5, NMI	
1	1	1	Level 0 to Level 6, NMI	Low

■ Interrupt Enable Flag (IE)

Interrupt enable flag (IE) enables/disables acceptance of maskable interrupts by the CPU’s internal interrupt acceptance circuit. A ‘1’ enables maskable interrupts; a ‘0’ disables maskable interrupts.

After the CPU accept interrupts, IE comes ‘0’ automatically and disables all interrupts until execution of RTI instruction. In the multi interrupt process, IE must be ‘1’ again by the program. In this case, analize the interrupt factor and clear the request at first.

■ Software Bits (S0, S1)

Software bits (S0, S1) is software control bit for operating sysutem (OS). It cannot be used from the general user program in the system using OS.

■ Saturation Operation Flag (ST)

It specifies the presence of add-subtract saturation process. All add-subtract instruction (ADD, ADDC, SUB, SUBC) are processed as follows:

ST = 0, or ST = 1 and the last operation instruction is PXST :

Carry out an operation as usual

ST = 1, or ST = 0 and the last operation instruction is PXST :

V = 1 and N = 1 -> Set the operation result to the maximum value of positive (x’007FFF’).

V = 1 and N = 0 -> Set the operation result to the maximum value of negative (x’FF8000’).

2-4 Special Function Registers

The MN102H74G/74F/74D/F74G locates peripheral function registers on memory (x'00FC00' to x'00FFFF') using memory mapped I/O method. Special function registers control peripheral circuits and CPU.

Table 2-4-1 List of Special Function Registers

Regiater	Address	R/W	Function	Pages
CPUM	x'00FC00'	R/W*1	CPU Mode Control Register	This Chapter
EFCR	x'00FC08'	R/W*1	Expansion Function Control Register	
MEMMD0	x'00FC30'	R/W*1	External Memory Mode Register 0	Chapter 3 Bus Interface
MEMMD1	x'00FC32'	R/W*1	External Memory Mode Register 1	
MEMMD2	x'00FC34'	R/W*1	External Memory Mode Register 2	
MEMMD3	x'00FC36'	R/W*1	External Memory Mode Register 3	
MEMMD0S	x'00FC38'	R/W*1	External Memory Mode Register 0S	
MEMMD1S	x'00FC3A'	R/W*1	External Memory Mode Register 1S	
MEMMD2S	x'00FC3C'	R/W*1	External Memory Mode Register 2S	
MEMMD3S	x'00FC3E'	R/W*1	External Memory Mode Register 3S	
IAGR	x'00FC0E'	R	Interrupt Accept Group Number Register	Chapter 4 Interrupts
G0ICR	x'00FC40'	R/W*1	Nonmaskable Interrupt Control Register	
G1ICR	x'00FC42'	R/W*1	Maskable Interrupt Control Register 1	
G2ICR	x'00FC44'	R/W*1	Maskable Interrupt Control Register 2	
G3ICR	x'00FC46'	R/W*1	Maskable Interrupt Control Register 3	
G4ICR	x'00FC48'	R/W*1	Maskable Interrupt Control Register 4	
G5ICR	x'00FC4A'	R/W*1	Maskable Interrupt Control Register 5	
G6ICR	x'00FC4C'	R/W*1	Maskable Interrupt Control Register 6	
G7ICR	x'00FC4E'	R/W*1	Maskable Interrupt Control Register 7	
G8ICR	x'00FC50'	R/W*1	Maskable Interrupt Control Register 8	
G9ICR	x'00FC52'	R/W*1	Maskable Interrupt Control Register 9	
EXTMD0	x'00FC56'	R/W*1	External Interrupt Edge Register 0	
EXTMD1	x'00FC58'	R/W*1	External Interrupt Edge Register 1	
WDREG	x'00FC5A'	R/W*1	Wacthdog Interrupt Expansion Control Register	
TM0BC	x'00FE00'	R	Timer 0 Binary Counter	Chapter 5 Timers
TM1BC	x'00FE01'	R	Timer 1 Binary Counter	
TM0BR	x'00FE10'	R/W	Timer 0 Base Register	
TM1BR	x'00FE11'	R/W	Timer 1 Base Register	
TM0MD	x'00FE20'	R/W*1	Timer 0 Mode Register	
TM1MD	x'00FE21'	R/W*1	Timer 1 Mode Register	
PS0BC	x'00FE08'	R	Prescaler 0 Binary Counter	
PS0BR	x'00FE18'	R	Prescaler 0 Base Register	
PS0MD	x'00FE28'	R/W*1	Prescaler 0 Mode Register	

*1 Part of the register is only readable.

Areas other than above special function registers are reserved.

Operation cannot be guaranteed when byte-accessing CPUM (x'00FC00') and EFCR (x'00FC08'). Word-access CPUM and EFCR always.

2-5 Standby Function

2-5-1 Overview

The MN102H74G/74F/74D/F74G has two system clock oscillation pins (high-speed oscillation pin and low-speed oscillation pin). In addition, this LSI has two CPU operating modes (NORMAL mode and SLOW mode) and two standby modes (HALT mode and STOP mode). Using these mode effectively reduces the power consumption.

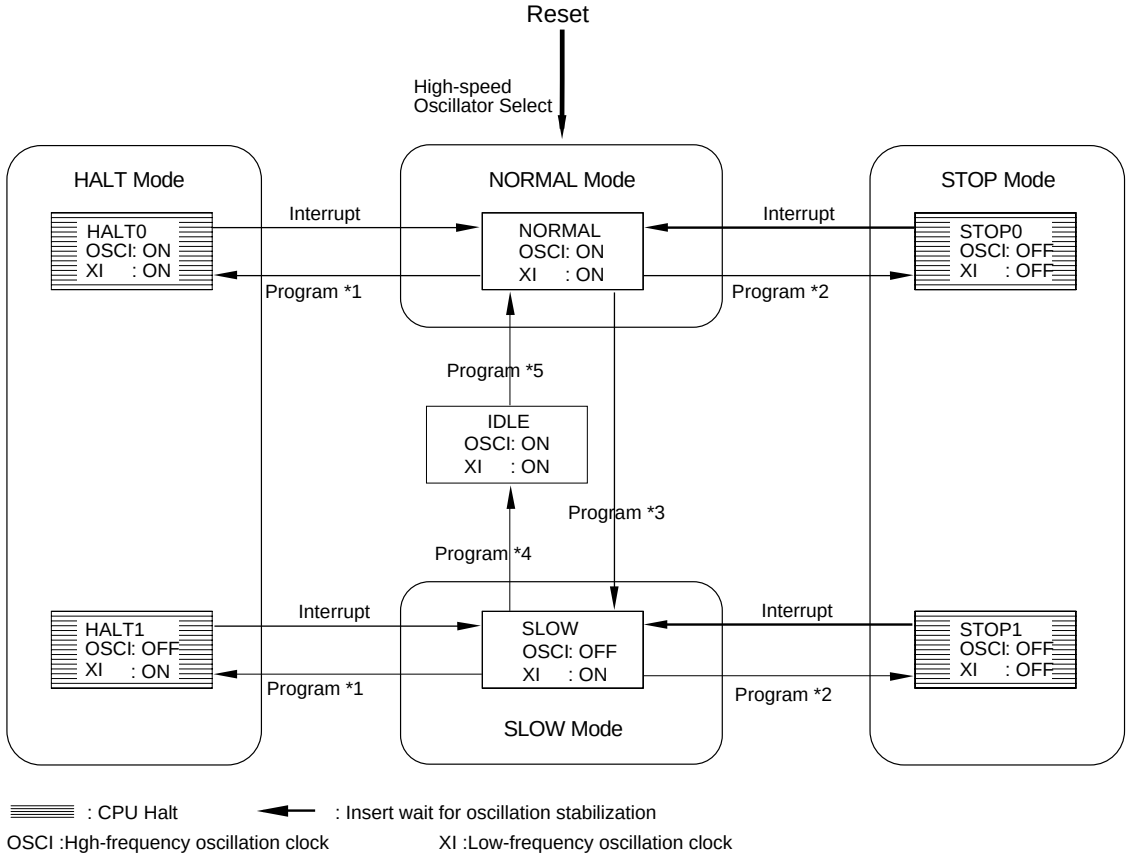


Figure 2-5-1 Transition Between Operation Modes

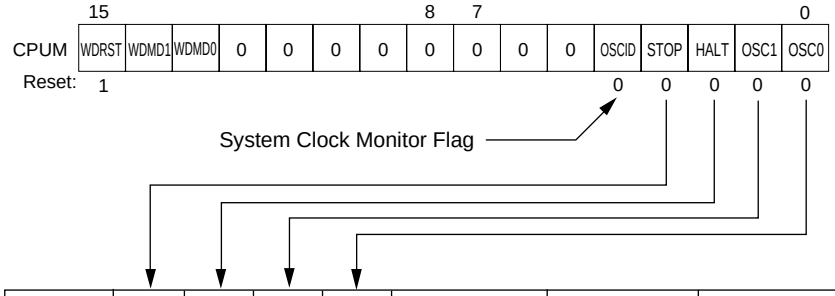
Examples of program *1 to program *5 are described in next section.

To reduce the power consumption in STOP mode and HALT mode, it is necessary to check the stability the output current from pins and the level of input pins. For output pins, the level should be the external level and the direction should be set to input. For input pins, the level should be fixed to the external level.

The MN102H74G/74F/74D/F74G has two system clock oscillation circuits. OSCI is a pin for high-speed oscillation (NORMAL mode) while XI is a pin for low-speed oscillation (SLOW mode). The CPU mode control register (CPUM) specifies the transfer between NORMAL mode and SLOW mode or between NORMAL/SLOW mode and standby mode. The normal reset operation and interrupts make the CPU return from standby mode. A wait for oscillation stability is inserted at reset or when returning from STOP mode, but is not inserted when returning from HALT mode. NORMAL/SLOW mode is automatically returned to the mode before the standby mode. Refer to “2-7 Reset Function”.

2-5-2 CPU Mode Control Register

The CPU mode control register controls transfers to each mode by setting associated flags.



Mode	STOP	HALT	OSC1	OSC0	High-frequency Clock	Low-frequency Clock	System Clock	CPU
NORMAL	0	0	0	0	Oscillation	Oscillation	OSCI	Operating
IDLE	0	0	0	1	Oscillation	Oscillation	XI	Operating
SLOW	0	0	1	1	Halt	Oscillation	XI	Operating
HALT0	0	1	0	0	Oscillation	Oscillation	OSCI	Halt
HALT1	0	1	1	1	Halt	Oscillation	XI	Halt
STOP0	1	0	0	0	Halt	Halt	Stop	Halt
STOP1	1	0	1	1	Halt	Halt	Stop	Halt

Figure 2-5-2 Operation Mode Control and Clock Oscillation

The following is the setup procedures to transfer from NORMAL mode to HALT mode or STOP mode.

(1) Clear the interrupt request flag (IQn) of the maskable interrupt control register (GnICR). Set the interrupt enable flag (IEn) and IE of PSW.

(2) Set CPUM to HALT mode or STOP mode.

Set the ILVn flag of ICRn as ILVn < IMn.

2-5-3 Transfer between NORMAL Mode and SLOW Mode

The MN102H74G/74F/74D/F74G has two CPU operating modes of NORMAL mode and SLOW mode. The CPU needs to go through idle mode once when transferring from SLOW mode to NORMAL mode.

The system clock monitor flag (OSCID) finds out whether the current system clock supplying to peripheral functions is high-speed oscillation clock or low-speed oscillation clock. OSCID is 0 means that the system clock is the high-speed oscillation clock, while OSCID is 1 means that the system clock is the low-speed oscillation clock.

The following is the program example to transfer from NORMAL mode to SLOW mode.

```
Program *3
            MOV   x'FC00', A1
            MOV   (A1), D0       ; Read CPUM
            OR    x'3', D0       ; Set SLOW mode
            MOV   D0, (A1)
```

Writing only the CPU mode control register sets the transfer from NORMAL mode to SLOW mode when the low-speed oscillation clock is stable enough. In this case, going though idle mode is not required.

The program needs to wait in idle mode until the high-speed oscillation clock starts and becomes stable enough when the CPU transfers from SLOW mode to NORMAL mode.

The CPU operates based on the low-speed oscillation clock in idle mode.

The wait time for oscillation stability requires the same time as reset. The program needs to count the time.

The following is the program example to transfer from SLOW mode to NORMAL mode.

```
Program *4
            MOV   x'FC00', A1
            MOV   (A1), D0        ; Read CPUM
            OR    x'FFFD', D0     ; Set IDLE mode
            MOV   D0, (A1)

Program *5
            MOV   35, D0          ; A loop to wait for approximately 5.5 ms with 32-kHz
LOOP        ADD   -1, D0          ; operation when switching from low-speed clock (32
            BNE   LOOP            ; kHz) to high-speed clock (12 MHz)
            MOV   x'FC00', A1     ; No need when this program continues from the program *4
            MOV   (A1), D0        ; Read CPUM
            AND   x'FFF0', D0     ; Set NORMAL mode
            MOV   D0, (A1)
```

2-5-4 Transfer to Standby Mode

The CPU switches from the CPU operating mode to standby mode using program and returns from standby mode to the CPU operating mode by an interrupt.

The following procedures require before switching to standby mode.

(1) Clear the interrupt enable flag (IE) of the processor status word (PSW) and the interrupt enable flag (IENn) of the maskable interrupt control register (ICRn). Disable all interrupts once.

(2) Specify the interrupt factor to return from standby mode to the CPU operating mode. Set only the associated IENn and then set the IE flag of PSW.

> **!** If the interrupt is enabled but interrupt priority level of the interrupt to be used is not equal to or higher than the mask level in PSW before transition to HALT or STOP mode, it is impossible to return to CPU operation mode by maskable interrupt.

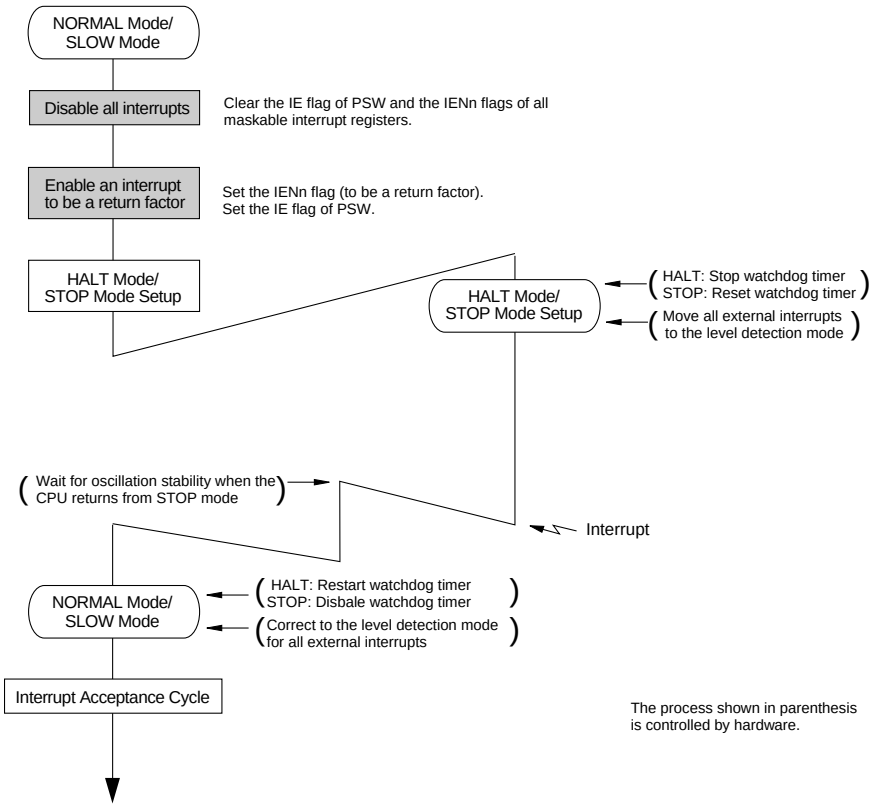

NORMAL Mode/
SLOW Mode
Disable all interrupts
Clear the IE flag of PSW and the IENn flags of all maskable interrupt registers.
Enable an interrupt to be a return factor
Set the IENn flag (to be a return factor).
Set the IE flag of PSW.
HALT Mode/
STOP Mode Setup
HALT Mode/
STOP Mode Setup
(HALT: Stop watchdog timer
STOP: Reset watchdog timer)
(Move all external interrupts to the level detection mode)
(Wait for oscillation stability when the CPU returns from STOP mode)
Interrupt
NORMAL Mode/
SLOW Mode
(HALT: Restart watchdog timer
STOP: Disbale watchdog timer)
(Correct to the level detection mode for all external interrupts)
Interrupt Acceptance Cycle
The process shown in parenthesis is controlled by hardware.

Figure 2-5-3 Sequence of Switching to/from Standby Mode

In the standby mode, all external interrupt detect modes becomes level detect. The rising edge detect is operated as 'H' level detect, and the falling edge detect is operated as 'L' level detect.

■ Transfer to HALT Mode

The CPU transfers from NORMAL mode to HALT0 mode and from SLOW mode to HALT1 mode. In both cases, only CPU stops keeping oscillation status. When the CPU switches to HALT mode while the watchdog timer is enabled, the watchdog timer stops counting. The following is the program example of switching to HALT mode.

```
Program *1
                mov     0xfc00 , a1
                mov     (a1), d0            ; Read CPUM
                or      0x4, d0             ; Set HALT mode
                jump    stp_hlt     ; Branch unconditionally to an even address
                align   2           ; to erase the difference of operating conditions.
stp_hlt         mov     d0, (a1)
                nop                 ; Insert more than three nops to execute a few
                nop                 ; instructions in the state of pipeline after
                nop                 ; writing to CPUM.
```

Assign the JMP instruction and set the CPUM write to an even address with ALIGN instruction. This prevents the effects due to the difference of memory mode and expansion bus widths and outcomes the same result under any conditions.

The ALIGN value must be set to more than 2 when the ALIGN value is set by the quasi-SECTION instruction before this example within the file describing the program.

■ Return from HALT Mode

An interrupt or a reset recovers the CPU from HALT mode. Reset proceeds normal operation. An interrupt returns the previous mode before entering HALT mode and the watchdog timer restarts counting.

■ Transfer to STOP Mode

The CPU transfers from NORMAL mode to STOP0 mode and from SLOW mode to STOP1 mode. In both cases, the oscillation and the CPU stop. When the CPU switches to STOP mode, the watchdog timer is reset. The following is the program example of switching to STOP mode.

```
Program *2
                mov     0xfc00 , a1
                mov     (a1), d0        ; Read CPUM
                or      0x8, d0         ; Set STOP mode
                jump    stp_hlt         ; Branch unconditionally to an even address
                align   2               ; to erase the difference of operating conditions.
stp_hlt         mov     d0, (a1)
                nop                     ; Insert more than three nops to execute a few
                nop                     ; instructions in the state of pipeline after
                nop                     ; writing to CPUM.
```

■ Return from STOP Mode

An interrupt or a reset recovers the CPU from STOP mode. At returning from STOP mode, the watchdog timer becomes disabled after operating as the oscillation stabilization wait counter.

When the CPU ends the oscillation stabilization wait and switches to the CPU operating mode, the watchdog timer becomes disabled automatically. When the watchdog timer operation is required, set the watchdog timer enabled.

The oscillation stabilization wait is executed by hardware when returning from STOP mode. The program does not need to count the oscillation stabilization wait time.

2-6 Error Detection Function

The MN102H74G/74F/74D/F74G has the error detection function as the interrupt vector allocated on the interrupt group 0. An error detection interrupt (nonmaskable interrupt) occurs based on the conditions shown in the following table.

Interrupt Trigger	Conditions	Interrupt Level
Error Detect Interrupt (Group 0)	When the undifined instruction is executed	Nonmaskable
	When the wacthdog timer overflows	Nonmaskable
	When the interrupt level which is notified in the maskable interrupt does not match the interrupt trigger	IAGR is 0 in nomaskable

Note: Reset IRQn in error before read IAGR during the interrupt preprocessing.

The CPU accepts an error detect interrupt with highest priority because the error detect interrupt is a nonmaskable interrupt. Therefore, the CPU detects errors during interrupt service routine.

The watchdog timer overflow interrupt is used to detect the CPU errors. The CPU cannot return to the previous operation before the watchdog timer overflow interrupt occurred after interrupt service routine is executed. Program the initialization program execution or error detect program in the watchdog timer overflow interrupt routine. Use a timer interrupt when the CPU needs to return to the previous operation. The NMIF, WDIF and UNIF remain '0'. In addition, use software as needed.

The watchdog timer is a 17-bit binary counter for the CPU base oscillation clock. The watchdog timer enable flag (WDRST) of the CPU mode control register (CPUM) controls the watchdog timer. Setting WDRST to '0' enables the watchdog timer while setting WDRST to '1' disables the watchdog timer. In addition, WDRST is '1' at reset. Set the watchdog timer overflow interrupt request flag (WDIF) of the nonmaskable interrupt control register (G0ICR) when the watchdog timer overflows. Set the undefined instruction interrupt request flag (UNIF) of the G0ICR register if an interrupt occurs by executing the undefined instruction. Specify its interrupt vector and its interrupt group number (IAGR) when a maskable interrupt occurs. The IAGR, however, is x'0000' when the interrupt level does not match the interrupt vector.

Setting WDRST to '0' immediately after setting this flag to '1' resets the watchdog timer. The watchdog timer must be reset within 2^{16} (65536) cycles by program to avoid the watchdog timer overflow.

2-7 Reset Function

Pulling the NRST pin low resets the CPU and initializes registers.

> The NRST must be low during 4 cycles or more of the base oscillation clock.

(1) Pulling the NRST pin low moves to reset mode.

> The chip model where the reset pin is shared with a port moves to reset mode by setting the port to '0'.

(2) Changing the NRST pin from low to high starts counting the base oscillation clock by the 17-bit binary counter. The interval from the counter starts until the counter counts up is the wait for oscillation stability. The reset releases by ending the wait for the oscillation stability.

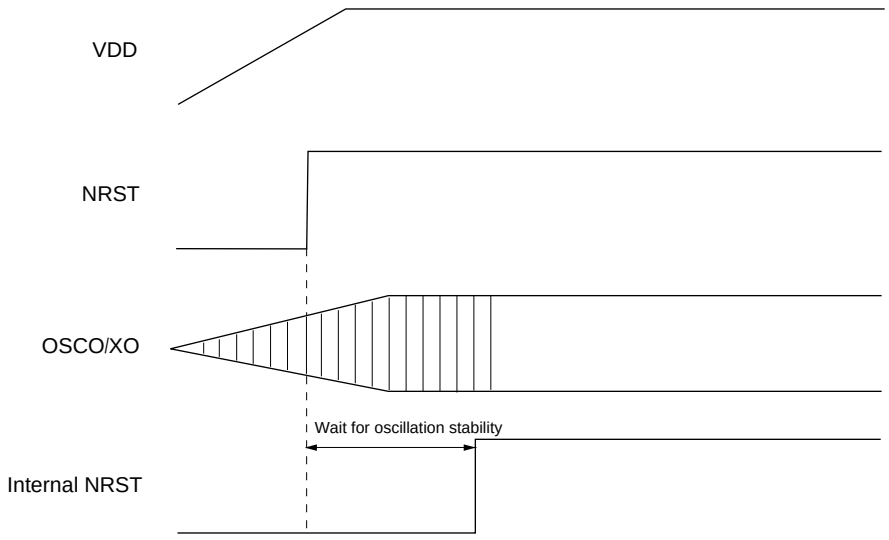


Figure 2-7-1 Sequence of Reset Release

> When connecting the NRST pin to the circuit for power level detection, use the circuit providing enough pulse being low level. Reset may start even if the NRST pin is low level for four cycles or less.

Calculate the wait for oscillation stability as follows.

High-speed oscillation:

tosciw = 2^{17} × (1/(2 × fosci))

For example, tosciw = 5.4613 ms when fosci = 12 MHz

Low-speed oscillation:

txiw = 2^{17} × (1 × fxi)

For example, txiw = 4.096 s when fxi = 32 kHz

(3) Initialize the internal registers and special registers as follows by hardware reset process.

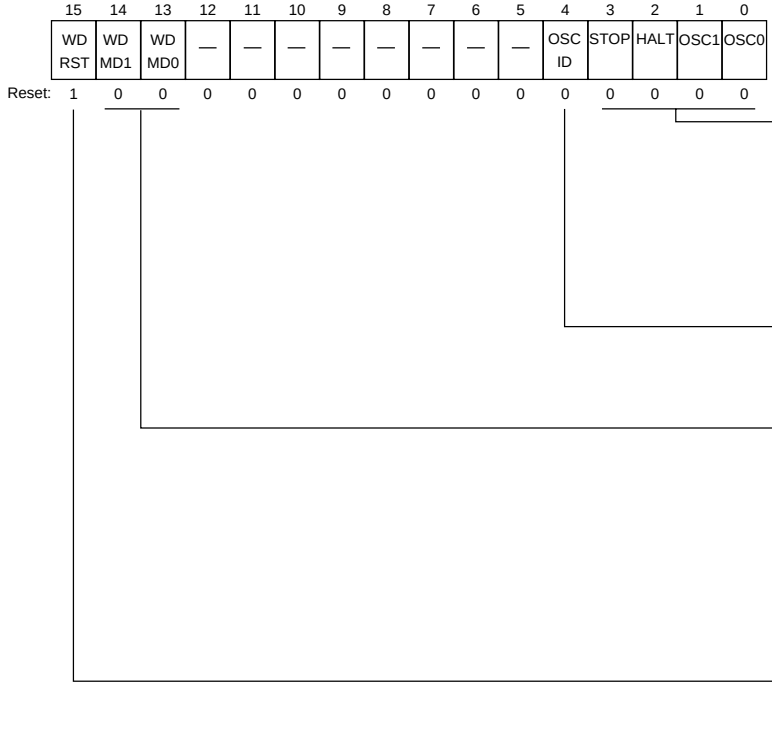
Table 2-7-1 Initial Value of Register at Reset

Register		Initial Value
Processor Status Word	PSW	x'0000'
Program Counter	PC	x'080000'
Address Register	An	Undefined
Data Register	Dn	Undefined
CPU Mode Control Register	CPUM	x'8000'
Expansion Function Control Register	EFCR	x'0000'
Interrupt Accept Group Number Register	IAGR	x'0000'
Interrupt Control Register	GnICR	x'0000'

(4) When the wait for the oscillation stability ends, the reset releases and the CPU starts executing the instruction from x'080000'. Allocate the initialization program on x'080000'. Allocate the JMP instruction on x'080000' since the entry address for interrupt is allocated on x'080008'.

(5) The CPU operates in the 7-cycle fixed wait mode immediately after reset process. Set the number of waits and bus mode for each system on the initialization program. (See "Chapter 3 Bus Interface")

2-8 CPU Special Register



STOP	HALT	OSC1	OSC0	Operating Mode	Oscillation	Clock	OSCID
0	0	0	0	NORMAL	ON	OSCI	0
0	0	1	1	SLOW	ON	SLOW	1
0	1	0	0	HALT0	ON	OSCI	0
0	1	1	1	HALT1	ON	SLOW	1
1	0	0	0	STOP0	OFF	-	-
1	0	1	1	STOP1	OFF	-	-

OSCID	System Clock Monitor Flag
0	OSCI input
1	Low-speed clock input

WDMD1	WDMD0	Watchdog Interrupt Cycle(BOSC Signal)	Return Time from STOP Mode
0	0	2^{17}Cycles	2^{17} × (1/fosci)
0	1	Reserved	Reserved
1	0	2^{13}Cycles	2^{13} × (1/fosci)
1	1	2^{15}Cycles	2^{15} × (1/fosci)

The WDMD1 and WDMD0 flags set the watchdog interrupt cycles and the rerutn time from stop mode. The set values are the same interval or time because the same counter is used. The WDREG register is set to extend the watchdog timer interrupt cycles.
Do not set '01' in the MN102H74G/74F/74D/F74G.

WDRST	Watchdog Timer Enable Flag
0	Enable
1	Disable and clear the watchdog timer

Figure 2-8-1 CPU Mode Control Register (CPUM: x'00FC00', R/W)

The following describes programming rules and precautions in the STOP/HALT mode.

Points for Programming

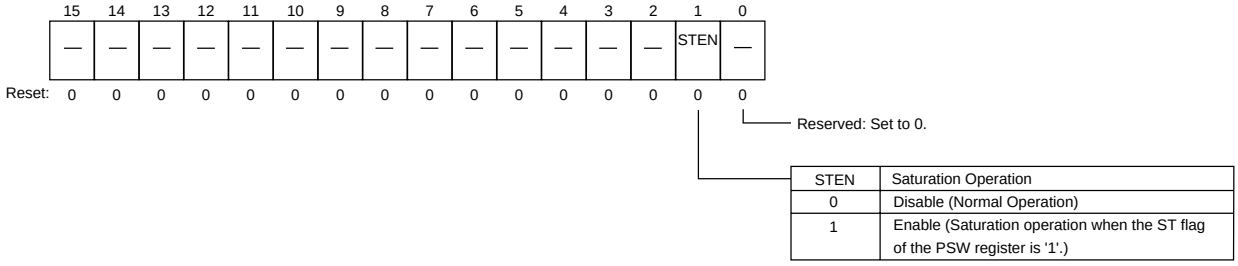
(1) Setting the CPUM address in the address register in advance, set the CPUM register using the MOV instruction with the register indirect addressing mode.
(2) Immediately after the MOV instruction, locate three NOPs consecutively.
(3) Immediately before the MOV instruction, locate the JMP instruction and align to the even address. This avoids the effects by the differences of the bus widths in the memory mode or expansion mode and provides the same result when operating in any conditions.

Programming Coding Example in Assembler (as102Ver.1.0, Ver.2.0)

```
            MOV     CPUM, A0    ; Set A0 to the CPUM address.
            MOV     (A0), D0    ; Transfer the contents of CPUM to D0.
            OR      x'000*', D0 ; Generate the data to set the STOP/HALT mode.
            JMP     STP_HLT     ; Branch unconditionally to the even address to
            ALIGN   2           ; eliminate the difference of operating conditions.
STP_HLT     MOV     D0, (A0)    ; Set the STOP/HALT mode to CPUM.
            NOP                 ; Dummy
            NOP                 ; Dummy
            NOP                 ; Dummy
```

Precautions

(1) * of OR instruction varies depending on the STOP or HALT mode.
(2) Set the ALIGN value to '2' or more in the above file when the ALIGN value is set using SECTION dummy instruction before this programming coding is described.
(3) Code the above programming in another file of the assembler source file when the program is developed with C complier cc102.



STEN	Saturation Operation
0	Disable (Normal Operation)
1	Enable (Saturation operation when the ST flag of the PSW register is '1'.)

Figure 2-8-2 Extension Function Control Register (EFCR: x'00FC08', R/W)

Chapter 3 Bus Interface

3-1 Memory

3-1-1 Internal Memory Location Examples

■ Processor Mode

The processor mode operates connecting to up to 16 MB external memory using Internal RAM and I/O ports.

The processor Mode uses the external memory, but does not use Internal ROM.

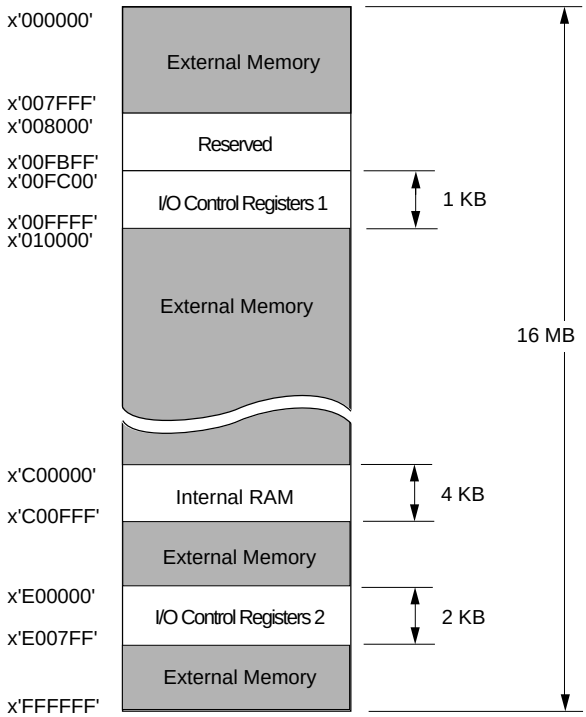


Figure 3-1-1 Address Space (Processor Mode)

The MN102H74G/74F/74D/F74G ignores accessing the external memory located on the 28 KB address space from x'008000' and accesses the reserved or I/O control register space in the chip.

The MN102H74G/74F/74D/F74G ignores accessing the external memory located on the 4 KB address space from x'C00000' and the 2 KB address space from x'E00000', and accesses the Internal RAM or I/O control register space in the chip.

The spaces for Internal ROM, Internal RAM and reserved in MN102H74G/74F/74D/F74G are different from spaces in the MN102L series.

■ Memory Expansion Mode (Single-chip Mode)

The memory expansion mode operates connecting to ROM, RAM and I/O ports externally. The memory expansion mode extends the memory externally using Internal ROM and Internal RAM.

The memory expansion mode uses up to 16 MB memory space. This mode is used when the program is larger than the Internal ROM capacity or the data is larger than the Internal RAM capacity. Accessing the external memory is not allowed when the system is configured with only internal memory in single-chip mode.

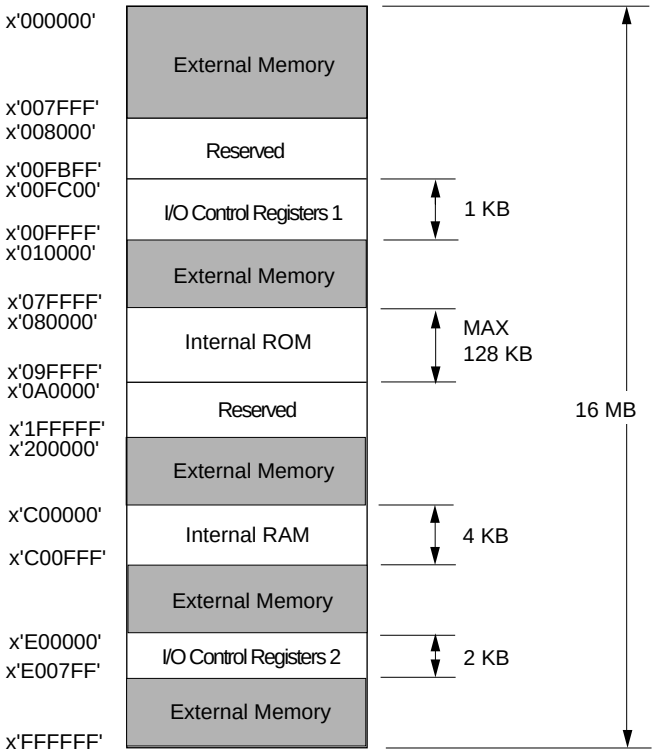


Figure 3-1-2 Address Space (Memory Expansion Mode)

The spaces for Internal ROM, Internal RAM and reserved in MN102H74G/74F/74D/F74G are different from spaces in the MN102L series.

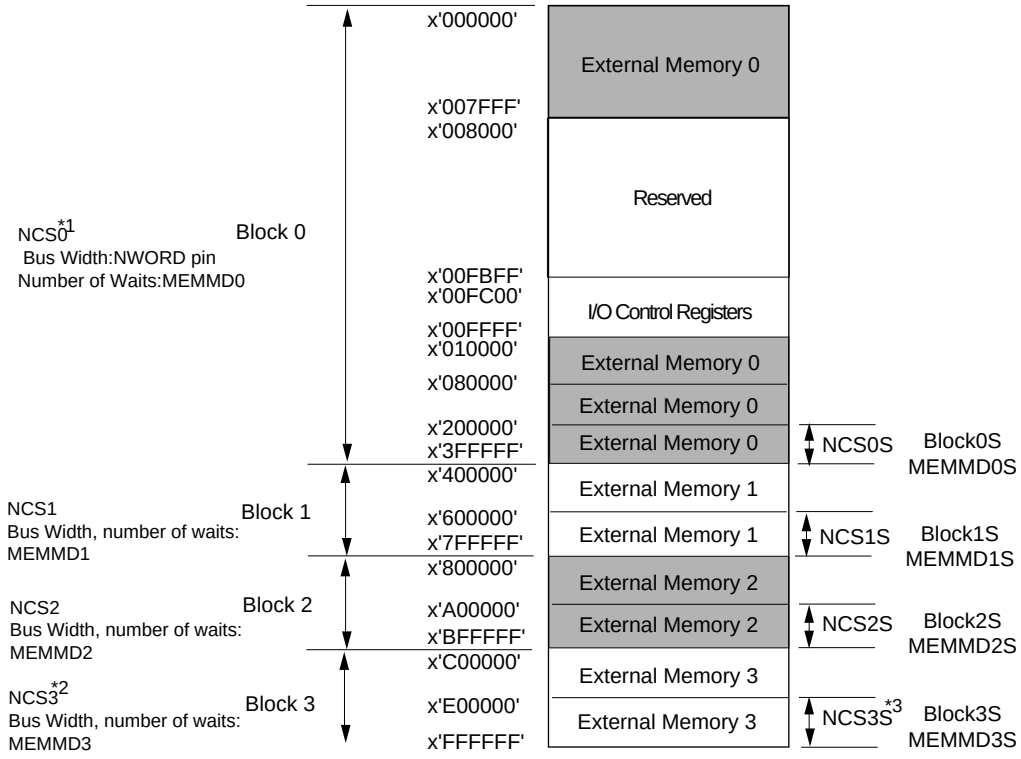
The value of internal RAM is uncertain when power is applied to it.
It needs to be initialized before used.

3-2 Summary of Bus Interface

3-2-1 Configuration

When the built-in chip-select function (NCS0 to NCS3, NCS0S to NCS3S) is used, the address space is divided into 8 fixed blocks (Block 0 to Block 3, Block 0S to Block 3S) and each block had approximately 2 MB area. (External address decoding is required when arbitrary chip-select functions are required.)

The 16-bit bus width or 8-bit bus width is selected for each block. The NWORD pin sets the bus width for Block 0 which contains a reset handler, while the MEMMDn registers or MEMMDnS registers set the bus width for Block 1 to Block 3 or Block 1S to Block 3S respectively.



*1 Except Internal ROM and I/O spaces.
*2 Except Internal RAM.
*3 Except I/O space.

Figure 3-2-1 Address Space

3-2-2 Control Registers

These registers control bus interface; the memory mode control register n (MEMMDn) and the memory mode control register nS (MEMMDnS).

Table 3-2-1 List of Bus Interface Control Registers

Registers	Address	R/W	Function
MEMMD0	x'00FC30'	R/W	External memory mode register 0
MEMMD1	x'00FC32'	R/W	External memory mode register 1
MEMMD2	x'00FC34'	R/W	External memory mode register 2
MEMMD3	x'00FC36'	R/W	External memory mode register 3
MEMMD0S	x'00FC38'	R/W	External memory mode register 0S
MEMMD1S	x'00FC3A'	R/W	External memory mode register 1S
MEMMD2S	x'00FC3C'	R/W	External memory mode register 2S
MEMMD3S	x'00FC3E'	R/W	External memory mode register 3S

The MEMMDn registers and MEMMDnS registers need to be set to match the system configuration. The MN102H74G/74F/74D/F74G selects one of three modes including normal mode, late access mode and WE short mode for each block. This LSI can directly connect to the memory with long hold time of read data or the external device required the hold time of write data.

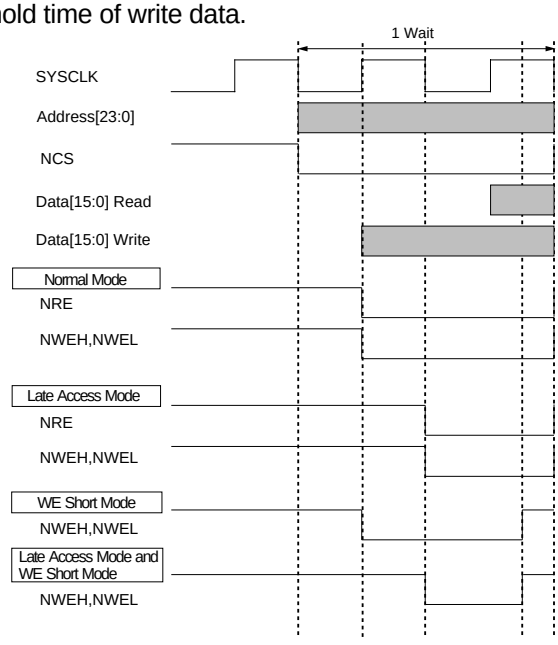


Figure 3-2-2 External Bus Access Signal

0 wait access is disabled in the external memory space.

Set NWEH and NWEL to short mode.

The MEMMDn(S) register sets the number of waits and bus mode for device connected to Blockn(S). The NWORD pin sets the bus width for Block 0. Therefore, the MEMMD0(S) register does not have the bus mode setup bit as other MEMMDn(S) registers have.
Set the MnSHRT flag and the MnLATE flag to '1' when using WE short mode and late access mode. In this case, do not use 0 wait access.
Set C-BUS mode for I/O control register 2 area. Set MnCBUS to '1'. Set MnSHRT and MnLATE to '0'.
Set MnHS to '1' when selecting the bus mode for Blockn(S) to handshake mode.

Bits[2:0] are ignored when handshake mode is selected.

When selecting fixed wait mode, set MnHS to '0' and MnWTm to the number of waits.

When setting MnHW to '1', 0.5 wait is added in the fixed wait mode.

Access is completed synchronizing with SYSCLK when no wait is selected in handshake mode. (SYSCLK falls at the beginning of the next access.)

Set MnHDBM only when 8-bit bus mode is selected.

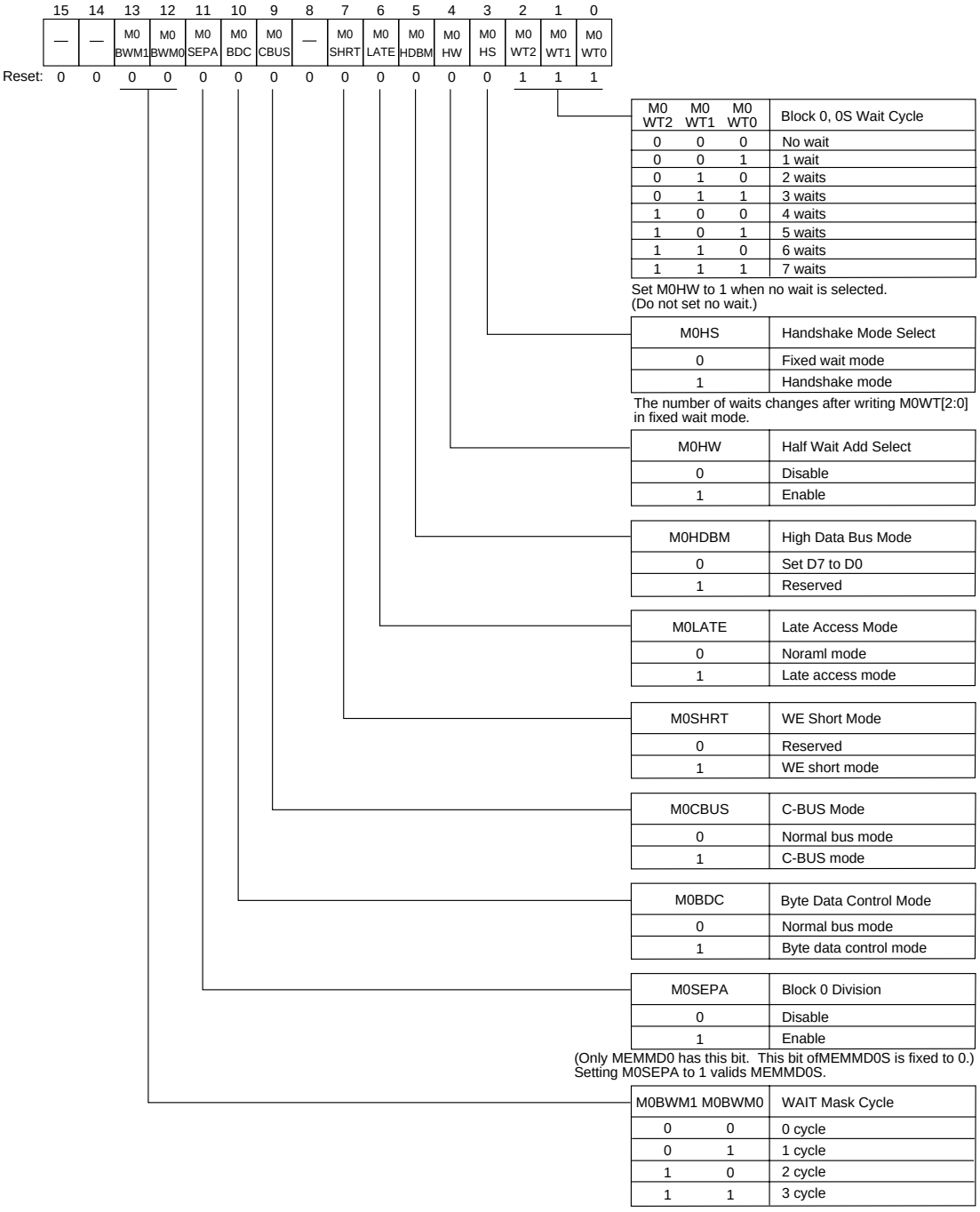

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
— — M0 BWM1 M0 BWM0 M0 SEPA M0 BDC M0 CBUS — M0 SHRT M0 LATE M0 HDBM M0 HW M0 HS M0 WT2 M0 WT1 M0 WT0
Reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
M0 WT2 M0 WT1 M0 WT0 Block 0, 0S Wait Cycle
0 0 0 No wait
0 0 1 1 wait
0 1 0 2 waits
0 1 1 3 waits
1 0 0 4 waits
1 0 1 5 waits
1 1 0 6 waits
1 1 1 7 waits
Set M0HW to 1 when no wait is selected.
(Do not set no wait.)
M0HS Handshake Mode Select
0 Fixed wait mode
1 Handshake mode
The number of waits changes after writing M0WT[2:0] in fixed wait mode.
M0HW Half Wait Add Select
0 Disable
1 Enable
M0HDBM High Data Bus Mode
0 Set D7 to D0
1 Reserved
M0LATE Late Access Mode
0 Noraml mode
1 Late access mode
M0SHRT WE Short Mode
0 Reserved
1 WE short mode
M0CBUS C-BUS Mode
0 Normal bus mode
1 C-BUS mode
M0BDC Byte Data Control Mode
0 Normal bus mode
1 Byte data control mode
M0SEPA Block 0 Division
0 Disable
1 Enable
(Only MEMMD0 has this bit. This bit ofMEMMD0S is fixed to 0.)
Setting M0SEPA to 1 valids MEMMD0S.
M0BWM1 M0BWM0 WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle

Figure 3-2-3 External Memory Mode Register 0 (MEMMD0) : x'00FC30'
External Memory Mode Register 0S (MEMMD0S) : x'00FC38'

Set bit 15, bit 14 and bit 8 to 0 always. Setup for unused block does not matter.

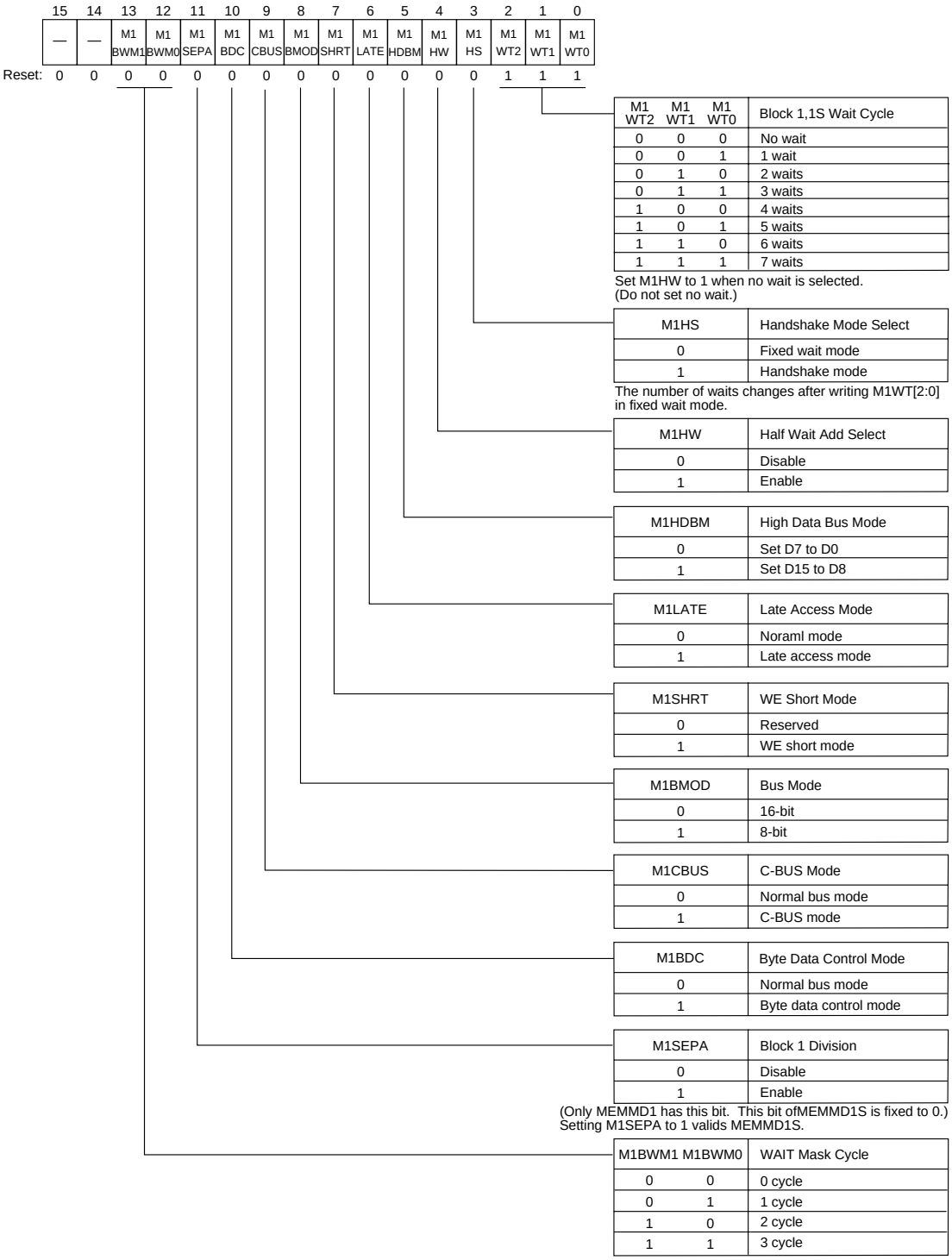

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
— — M1 BWM1 M1 BWM0 M1 SEPA M1 BDC M1 CBUS M1 BMOD M1 SHRT M1 LATE M1 HDBM M1 HW M1 HS M1 WT2 M1 WT1 M1 WT0
Reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
M1 WT2 M1 WT1 M1 WT0 Block 1,1S Wait Cycle
0 0 0 No wait
0 0 1 1 wait
0 1 0 2 waits
0 1 1 3 waits
1 0 0 4 waits
1 0 1 5 waits
1 1 0 6 waits
1 1 1 7 waits
Set M1HW to 1 when no wait is selected.
(Do not set no wait.)
M1HS Handshake Mode Select
0 Fixed wait mode
1 Handshake mode
The number of waits changes after writing M1WT[2:0] in fixed wait mode.
M1HW Half Wait Add Select
0 Disable
1 Enable
M1HDBM High Data Bus Mode
0 Set D7 to D0
1 Set D15 to D8
M1LATE Late Access Mode
0 Noraml mode
1 Late access mode
M1SHRT WE Short Mode
0 Reserved
1 WE short mode
M1BMOD Bus Mode
0 16-bit
1 8-bit
M1CBUS C-BUS Mode
0 Normal bus mode
1 C-BUS mode
M1BDC Byte Data Control Mode
0 Normal bus mode
1 Byte data control mode
M1SEPA Block 1 Division
0 Disable
1 Enable
(Only MEMMD1 has this bit. This bit ofMEMMD1S is fixed to 0.)
Setting M1SEPA to 1 valids MEMMD1S.
M1BWM1 M1BWM0 WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle

Figure 3-2-4 External Memory Mode Register 1 (MEMMD1) : x'00FC32'
External Memory Mode Register 1S (MEMMD1S) : x'00FC3A'

Set bit 15 and bit 14 to 0 always. Setup for unused block does not matter.

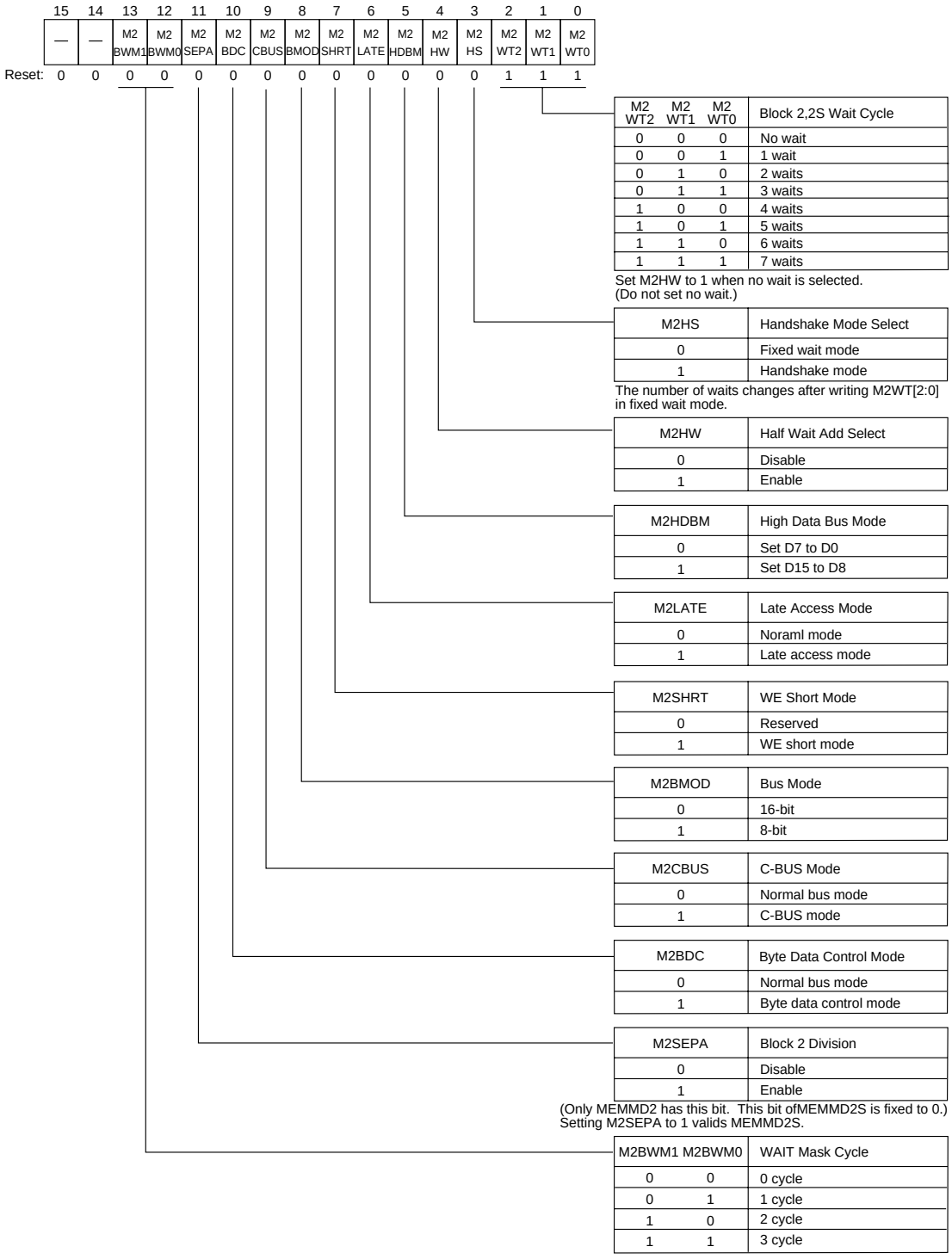

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
— — M2 BWM1 M2 BWM0 M2 SEPA M2 BDC M2 CBUS M2 BMOD M2 SHRT M2 LATE M2 HDBM M2 HW M2 HS M2 WT2 M2 WT1 M2 WT0
Reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
M2 WT2 M2 WT1 M2 WT0 Block 2,2S Wait Cycle
0 0 0 No wait
0 0 1 1 wait
0 1 0 2 waits
0 1 1 3 waits
1 0 0 4 waits
1 0 1 5 waits
1 1 0 6 waits
1 1 1 7 waits
Set M2HW to 1 when no wait is selected.
(Do not set no wait.)
M2HS Handshake Mode Select
0 Fixed wait mode
1 Handshake mode
The number of waits changes after writing M2WT[2:0] in fixed wait mode.
M2HW Half Wait Add Select
0 Disable
1 Enable
M2HDBM High Data Bus Mode
0 Set D7 to D0
1 Set D15 to D8
M2LATE Late Access Mode
0 Noraml mode
1 Late access mode
M2SHRT WE Short Mode
0 Reserved
1 WE short mode
M2BMOD Bus Mode
0 16-bit
1 8-bit
M2CBUS C-BUS Mode
0 Normal bus mode
1 C-BUS mode
M2BDC Byte Data Control Mode
0 Normal bus mode
1 Byte data control mode
M2SEPA Block 2 Division
0 Disable
1 Enable
(Only MEMMD2 has this bit. This bit ofMEMMD2S is fixed to 0.)
Setting M2SEPA to 1 valids MEMMD2S.
M2BWM1 M2BWM0 WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle

Figure 3-2-5 External Memory Mode Register 2 (MEMMD2) : x'00FC34'
External Memory Mode Register 2S (MEMMD2S) : x'00FC3C'

Set bit 15 and bit 14 to 0 always. Setup for unused block does not matter.

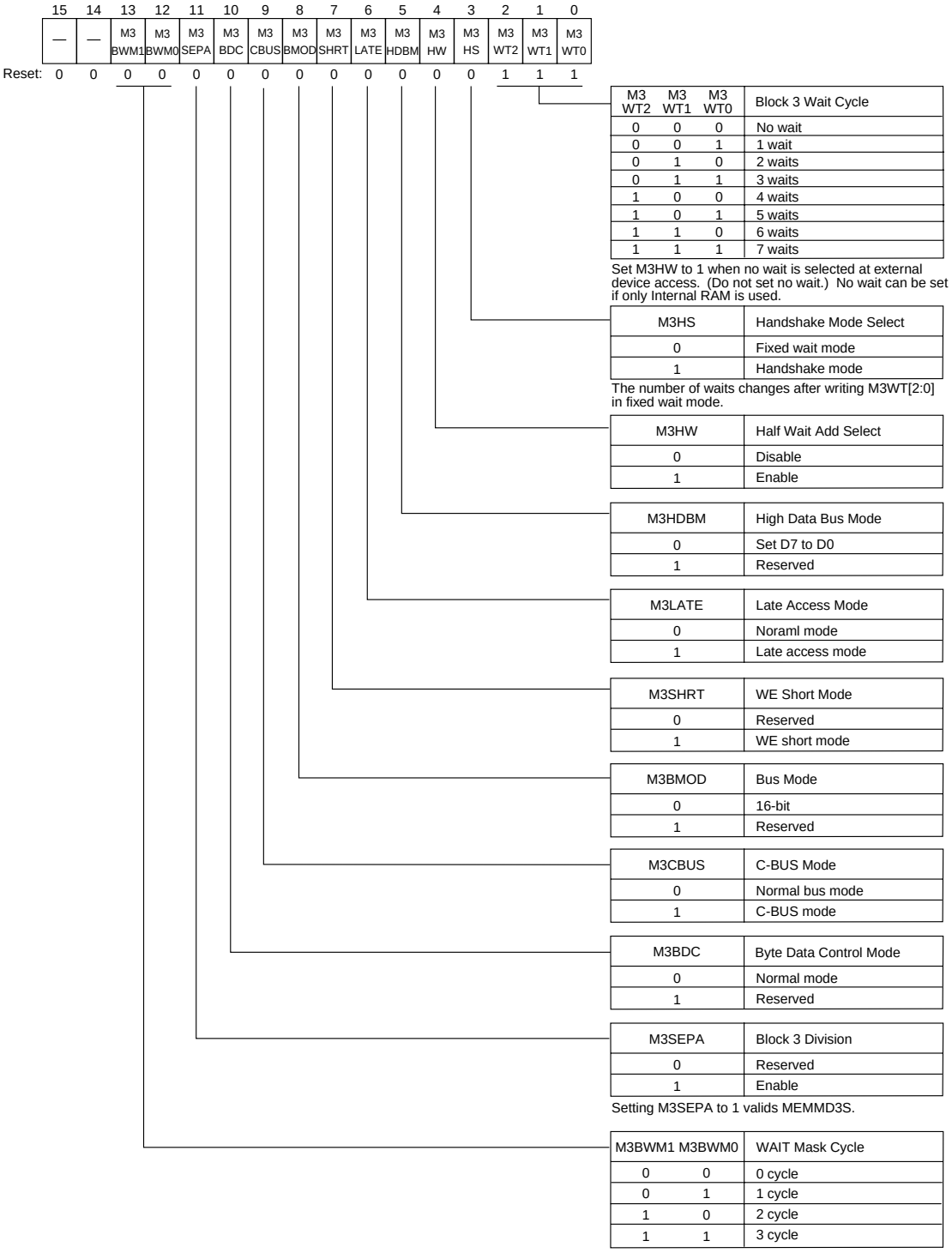

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
— — M3 BWM1 M3 BWM0 M3 SEPA M3 BDC M3 CBUS M3 BMOD M3 SHRT M3 LATE M3 HDBM M3 HW M3 HS M3 WT2 M3 WT1 M3 WT0
Reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
M3 WT2 M3 WT1 M3 WT0 Block 3 Wait Cycle
0 0 0 No wait
0 0 1 1 wait
0 1 0 2 waits
0 1 1 3 waits
1 0 0 4 waits
1 0 1 5 waits
1 1 0 6 waits
1 1 1 7 waits
Set M3HW to 1 when no wait is selected at external device access. (Do not set no wait.) No wait can be set if only Internal RAM is used.
M3HS Handshake Mode Select
0 Fixed wait mode
1 Handshake mode
The number of waits changes after writing M3WT[2:0] in fixed wait mode.
M3HW Half Wait Add Select
0 Disable
1 Enable
M3HDBM High Data Bus Mode
0 Set D7 to D0
1 Reserved
M3LATE Late Access Mode
0 Noraml mode
1 Late access mode
M3SHRT WE Short Mode
0 Reserved
1 WE short mode
M3BMOD Bus Mode
0 16-bit
1 Reserved
M3CBUS C-BUS Mode
0 Normal bus mode
1 C-BUS mode
M3BDC Byte Data Control Mode
0 Normal mode
1 Reserved
M3SEPA Block 3 Division
0 Reserved
1 Enable
Setting M3SEPA to 1 valids MEMMD3S.
M3BWM1 M3BWM0 WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle

Figure 3-2-6 External Memory Mode Register 3 (MEMMD3) : x'00FC36'

Set bit 15 and bit 14 to 0 always. Set MEMMD3 to internal RAM space access.

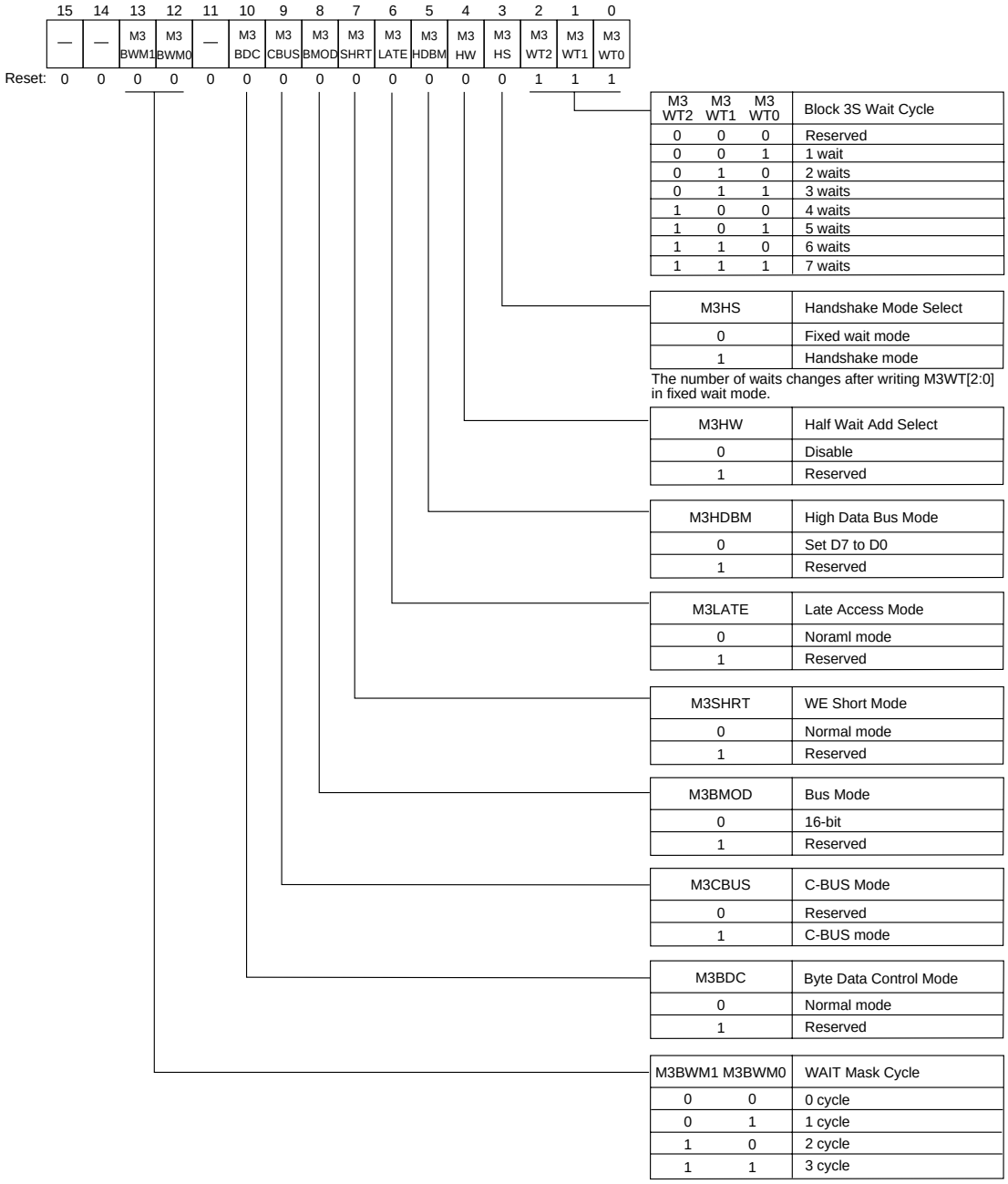

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
— — M3 BWM1 M3 BWM0 — M3 BDC M3 CBUS M3 BMOD M3 SHRT M3 LATE M3 HDBM M3 HW M3 HS M3 WT2 M3 WT1 M3 WT0
Reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
M3 WT2 M3 WT1 M3 WT0 Block 3S Wait Cycle
0 0 0 Reserved
0 0 1 1 wait
0 1 0 2 waits
0 1 1 3 waits
1 0 0 4 waits
1 0 1 5 waits
1 1 0 6 waits
1 1 1 7 waits
M3HS Handshake Mode Select
0 Fixed wait mode
1 Handshake mode
The number of waits changes after writing M3WT[2:0] in fixed wait mode.
M3HW Half Wait Add Select
0 Disable
1 Reserved
M3HDBM High Data Bus Mode
0 Set D7 to D0
1 Reserved
M3LATE Late Access Mode
0 Noraml mode
1 Reserved
M3SHRT WE Short Mode
0 Normal mode
1 Reserved
M3BMOD Bus Mode
0 16-bit
1 Reserved
M3CBUS C-BUS Mode
0 Reserved
1 C-BUS mode
M3BDC Byte Data Control Mode
0 Normal mode
1 Reserved
M3BWM1 M3BWM0 WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle

Figure 3-2-7 External Memory Mode Register 3S (MEMMD3S) : x'00FC3E'

Set bit 15 and bit 14 to 0 always. Set MEMMD3S to Internal I/O space access.

3-2-3 Transfer Bus Width Selection

A 8-bit data bus width or a 16-bit bus width is selected as the external extension bus width. The NWORD pin sets the bus width for Block 0. Pulling the NWORD pin high sets the 8-bit bus width, while pulling the NWORD pin low sets the 16-bit bus width. The MEMMD1 to 3 registers and the MEMMD1S to 3S registers sets the bus width for Block 1 to 3 and BLOCK 1S to 3S respectively. Refer to “3-2-2 Control Registers” for details.

When the 8-bit bus width is selected, the MnHDBM flag sets high data bus mode or low data bus mode. When MnHDBM is 1, only the upper 8-bit data (D15 to D8) and the write enable signal NWEH are valid. When MnHDBM is 0, only the lower 8-bit data (D7 to D0) and the write enable signal NWEL are valid. When the data is word-transferred in the 8-bit bus mode, the lower 8-bit data is transferred during 1st bus cycle while the upper 8-bit data is transferred during 2nd bus cycle. When the 24-bit bus data is loaded or stored, 3 bus cycles are generated consecutively. First, access the lower bytes during 1st bus cycle. Next, LSB of address adds by 1 and access the middle bytes during 2nd bus cycle. Finally, add the address by 2 and access the upper bytes during 3rd bus cycle.

Table 3-2-2 External Bus Access Mode

<table>
<tr><th>Mode</th><th>Data Size</th><th>NWEH</th><th>NWEL</th><th>1st Bus Cycle</th><th>2nd Bus Cycle</th><th>3rd Bus Cycle</th><th></th></tr>
<tr><td rowspan="3">8-bit</td><td>8-bit</td><td rowspan="3">H
(Low Data Bus Mode)
L
(High Data Bus Mode)</td><td rowspan="3">L
(Low Data Bus Mode)
H
(High Data Bus Mode)</td><td>(Valid)</td><td>---</td><td>---</td><td>Odd/Even Address Access Enable</td></tr>
<tr><td>16-bit</td><td>Lower Bytes
(Address LSB=0)</td><td>Upper Bytes
(Address LSB=1)</td><td>---</td><td rowspan="2">From Odd Address Access Disable</td></tr>
<tr><td>24-bit</td><td>Lower Bytes
(Address LSB=0)</td><td>Middle Bytes
(Address LSB=1)</td><td>Upper Bytes
(Address LSB=0, Address+2)</td></tr>
<tr><td rowspan="5">16-bit</td><td rowspan="2">8-bit</td><td>H</td><td>L</td><td>D7-0 Enable
(Address LSB=0)</td><td>---</td><td>---</td><td>When Even Address Access</td></tr>
<tr><td>L</td><td>H</td><td>D15-8 Enable
(Address LSB=1)</td><td>---</td><td>---</td><td>When Odd Address Access</td></tr>
<tr><td>16-bit</td><td>L</td><td>L</td><td>(Valid)
(Address LSB=0)</td><td>---</td><td>---</td><td rowspan="3">From Odd Address Access Disable</td></tr>
<tr><td rowspan="2">24-bit</td><td>L</td><td>L</td><td rowspan="2">Lower/Middle Bytes
(Address LSB=0)</td><td rowspan="2">Upper Bytes
(D7-0 Valid)
(Address LSB=0, Address+2)</td><td rowspan="2">---</td></tr>
<tr><td>H</td><td>L</td></tr>
</table>

3-3 External Bus Connection Examples

3-3-1 External Bus Connection Examples

This section describes external bus connection examples.

■ ROM Connection Using 16-bit bus width

The following example shows that two 256-KB ROM are allocated in Block 0. The MN102H74G/74F/74D/F74G starts x'080000' after reset release. Therefore, the space from x'080000' to x'0FFFFF' is valid. The ROM data is read in x'000000' to x'00EFFF', x'010000' to x'07FFFF', x'100000' to x'17FFFF', x'180000' to x'1FFFFF', x'200000' to x'27FFFF', x'280000' to x'2FFFFF', x'300000' to x'37FFFF' or x'380000' to x'3FFFFF'. When 16-bit data is word-transferred in two 256-KB ROM, the address is shifted by 1. For example, the address A1 in the MN102H74G/74F/74D/F74G is connected to the address A0 in ROM.

MEMMD0: x'00FC30'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	M0 BWM1	M0 BWM0	M0 SEPA	M0 BDC	M0 CBUS	-	M0 SHRT	M0 LATE	M0 HDBM	M0 HW	M0 HS	M0 WT2	M0 WT1	M0 WT0
-	-	-	-	0	-	0	-	1	0	-	1	0	0	0	0

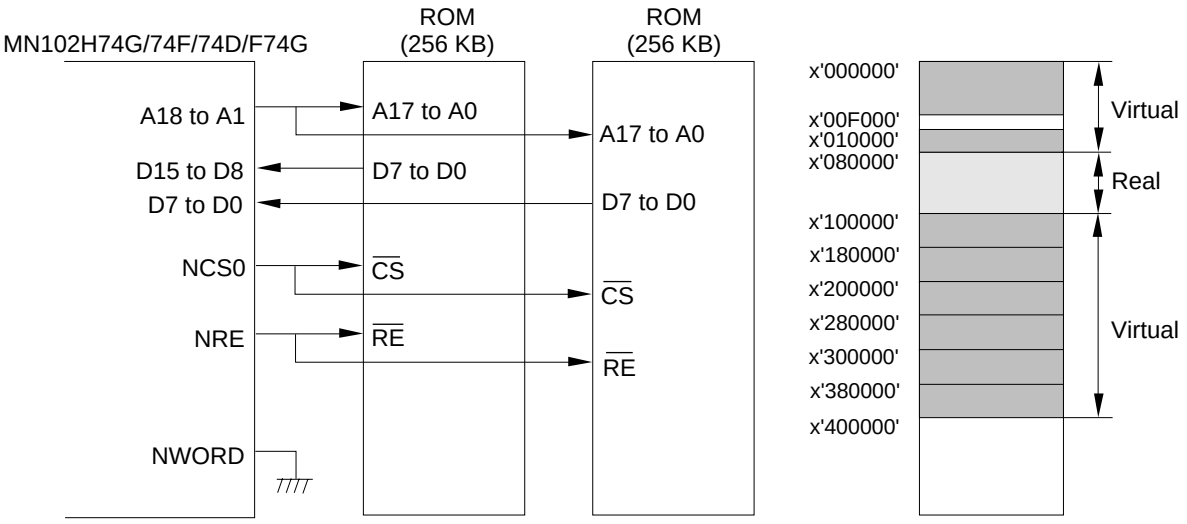


Figure 3-3-1 ROM Connection (16-bit Bus Width)

■ ROM Connection Using 8-bit bus width

The following example shows that 512-KB ROM is allocated in Block 0 in late access mode. The memory map is the same as one in 16-bit bus mode, but the connection method is a little different from the one in 16-bit bus mode. D7 to D0 are used.

MEMMD0: x'00FC30'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	M0 BWM1	M0 BWM0	M0 SEPA	M0 BDC	M0 CBUS	-	M0 SHRT	M0 LATE	M0 HDBM	M0 HW	M0 HS	M0 WT2	M0 WT1	M0 WT0
-	-	-	-	0	-	0	-	1	0	-	-	0	0	0	1

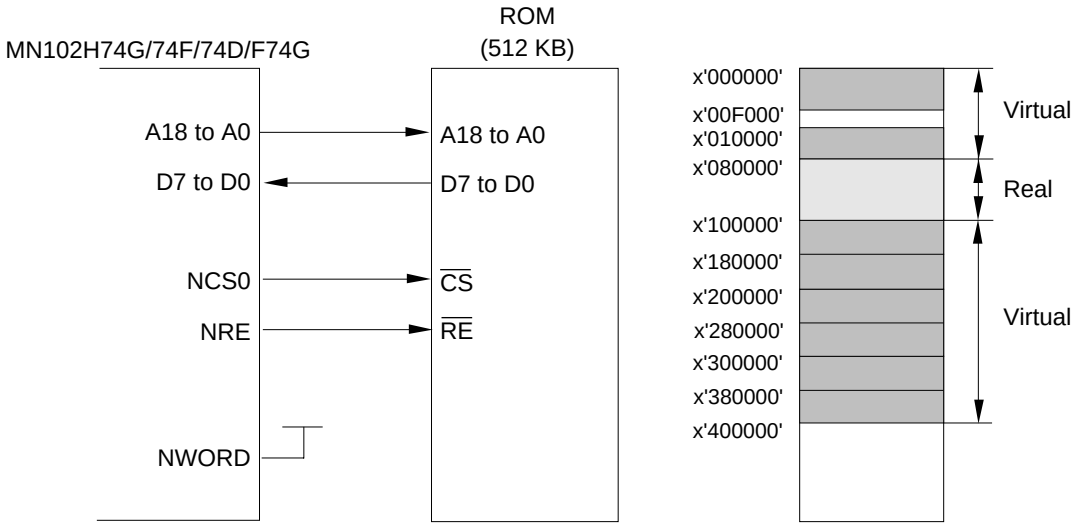


Figure 3-3-2 ROM Connection (8-bit Bus Width)

■ SRAM Connection Using 16-bit bus width

The following example shows that two 32-KB SRAMs are allocated in Block1. The area of x'400000' to x'40FFFF' is the real RAM space. The same RAM data is read in x'410000' to x'41FFFF' and x'7F0000' to x'7FFFFF' (These areas are virtual) due to the lack of address decoder. When 16-bit data is word-transferred, the address is shifted by 1. For example, the address A1 in the MN102H74G/74F/74D/F74G is connected to the address A0 in SRAM.

MEMMD1: x'00FC32'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	M1 BWM1	M1 BWM0	M1 SEPA	M1 BDC	M1 CBUS	M1 BMOD	M1 SHRT	M1 LATE	M1 HDBM	M1 HW	M1 HS	M1 WT2	M1 WT1	M1 WT0
-	-	-	-	0	-	0	0	1	0	-	-	0	0	0	1

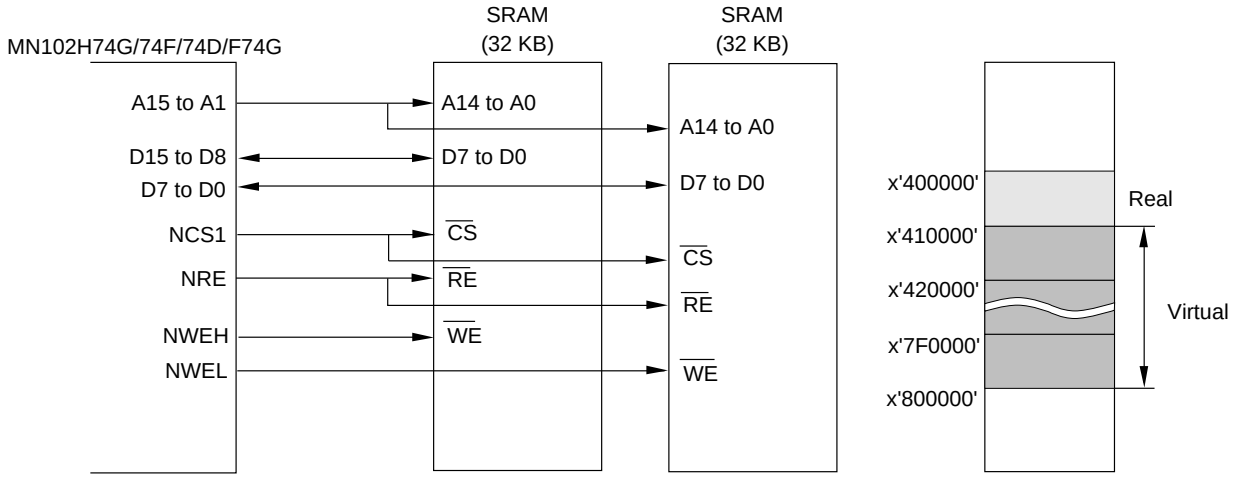


Figure 3-3-3 RAM Connection (16-bit Bus Width)

■ SRAM Connection Using 8-bit bus width

The following example shows that 64-KB SRAM is allocated in Block 1 in late access mode. The area of x'400000' to x'40FFFF' is the real RAM space. The same RAM data is read in x'410000' to x'41FFFF' and x'7F0000' to x'7FFFFF' (These areas are virtual) due to the lack of address decoder. D7 to D0 are used.

MEMMD1: x'00FC32'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	M1 BWM1	M1 BWM0	M1 SEPA	M1 BDC	M1 CBUS	M1 BMOD	M1 SHRT	M1 LATE	M1 HDBM	M1 HW	M1 HS	M1 WT2	M1 WT1	M1 WT0
-	-	-	-	0	-	0	1	1	1	-	-	0	0	0	1

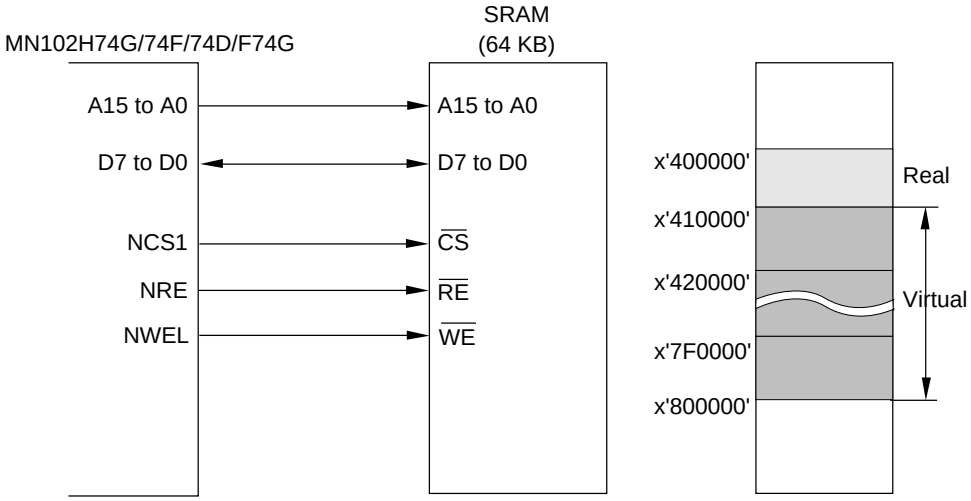


Figure 3-3-4 RAM Connection (8-bit Bus Width)

■ Byte Data Control Mode

When the byte data control mode flag is set to enable (the MnBDC flag is 1), NWEH, NWEL and NWR is selected to NUB, NLB and NWE respectively.

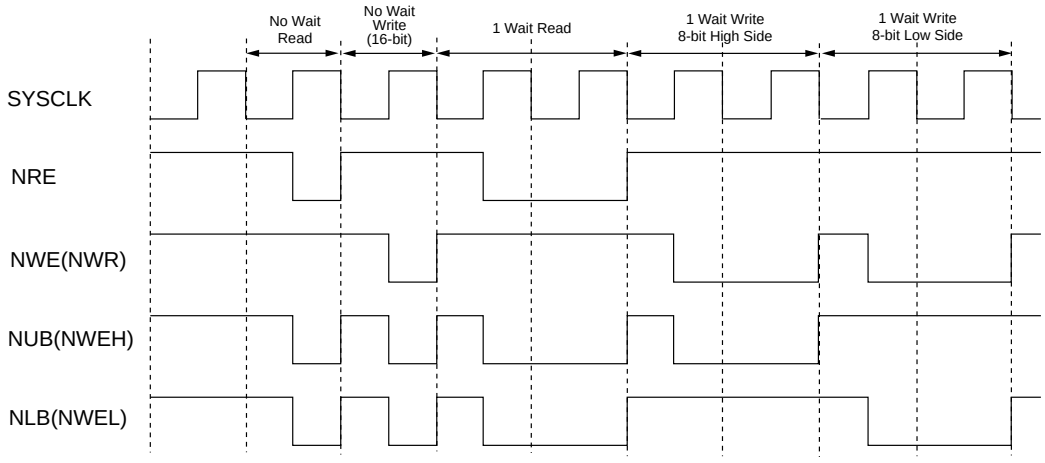


Figure 3-3-5 Byte Data Control Mode Access Timing

■ C-BUS Mode

The MN102H74G/74F/74D/F74G supports C-BUS mode configurated 2 cycles or more. Select C-BUS mode for the I/O control register 2 (See 3-2, 3-3) space.

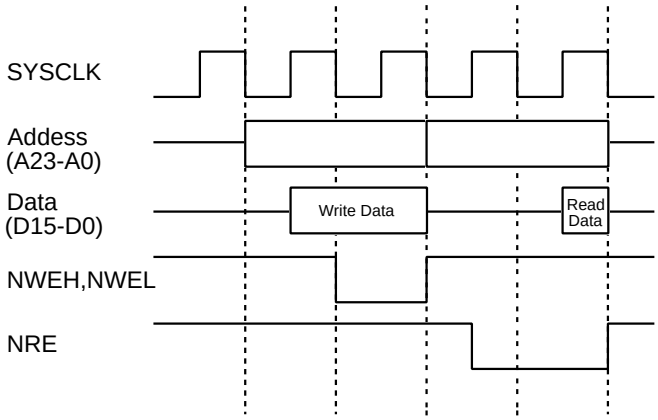


Figure 3-3-6 C-BUS Access Timing

Set the external memory mode register n (MEMMDn) as follows. Set MnWT[2:0] to 1 wait or more.

MEMMD1: x'00FC32'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	M1 BWM1	M1 BWM0	M1 SEPA	M1 BDC	M1 CBUS	M1 BMOD	M1 SHRT	M1 LATE	M1 HDBM	M1 HW	M1 HS	M1 WT2	M1 WT1	M1 WT0
-	-	-	-	0	-	1	-	0	0	-	-	0	0	0	1

■ External Bus Access Timing Using 16-bit Bus Mode (MnCBUS of MEMMDn =0)

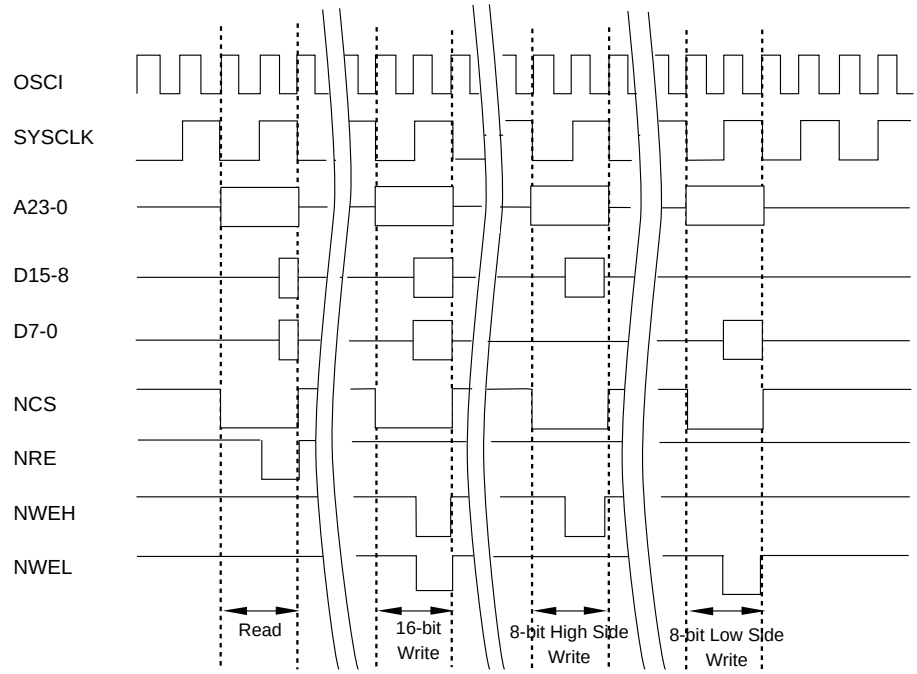


Figure 3-3-7 16-bit Bus Access Timing (No Wait)

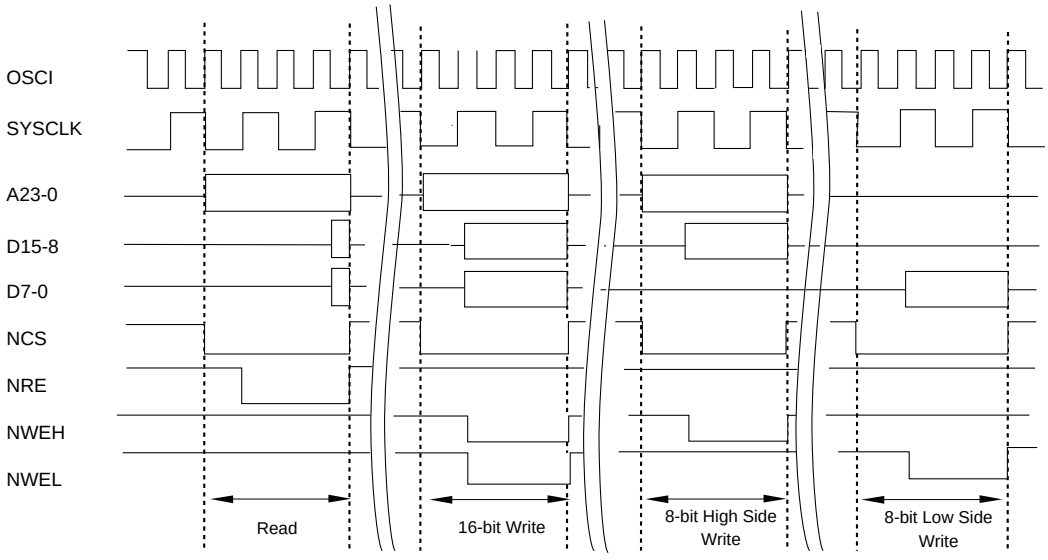


Figure 3-3-8 16-bit Bus Access Timing (1 Wait)

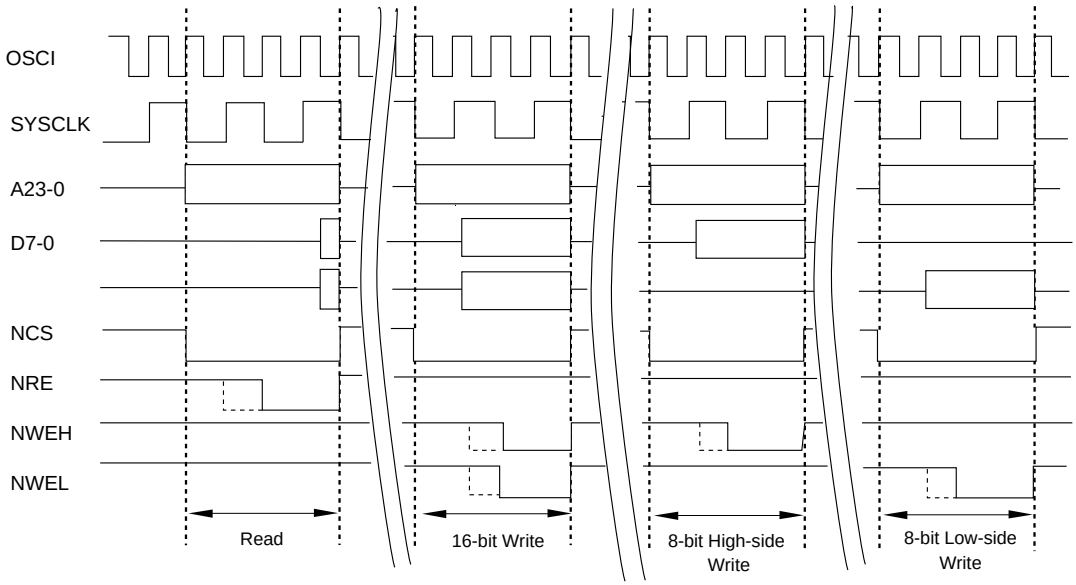


Figure 3-3-9 16-bit Bus Access Timing (1 Wait) (Late Access Mode)

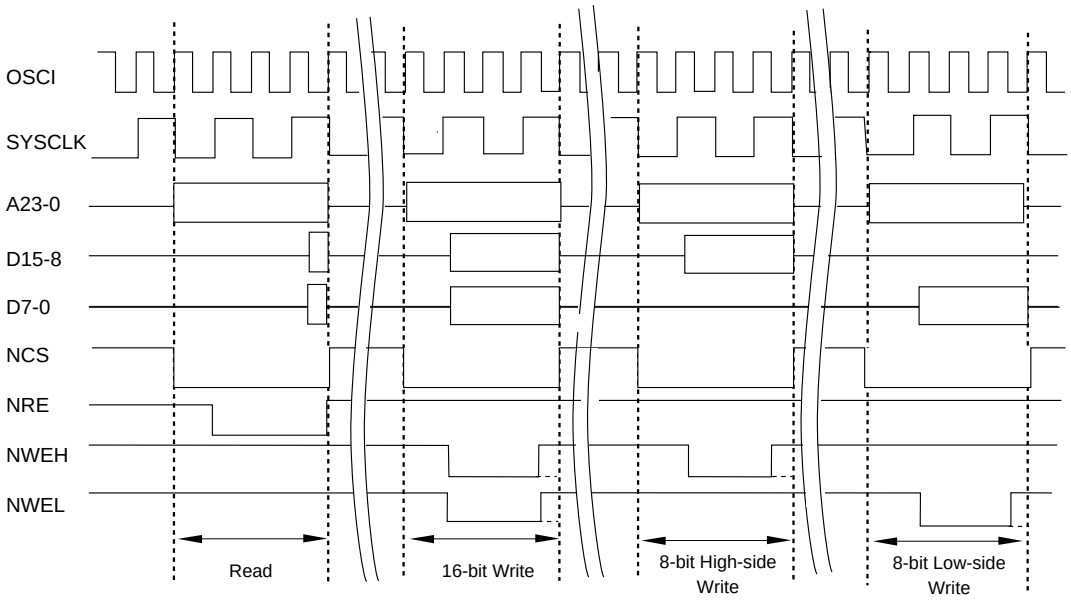


Figure 3-3-10 16-bit Bus Access Timing (1 Wait) (WE Short Mode)

■ External Bus Access Timing Using 8-bit Bus Mode (MnCBUS of MEMMDn =0)

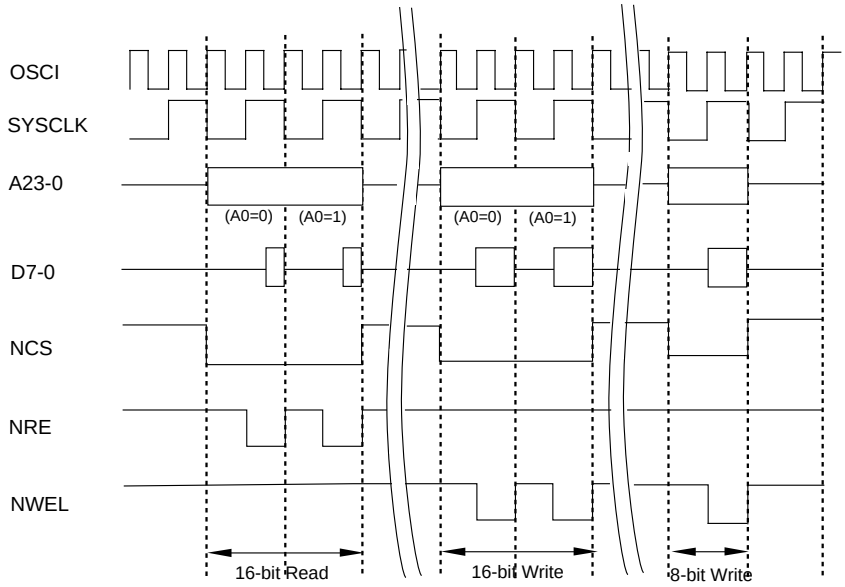


Figure 3-3-11 8-bit Bus Access Timing (No Wait)

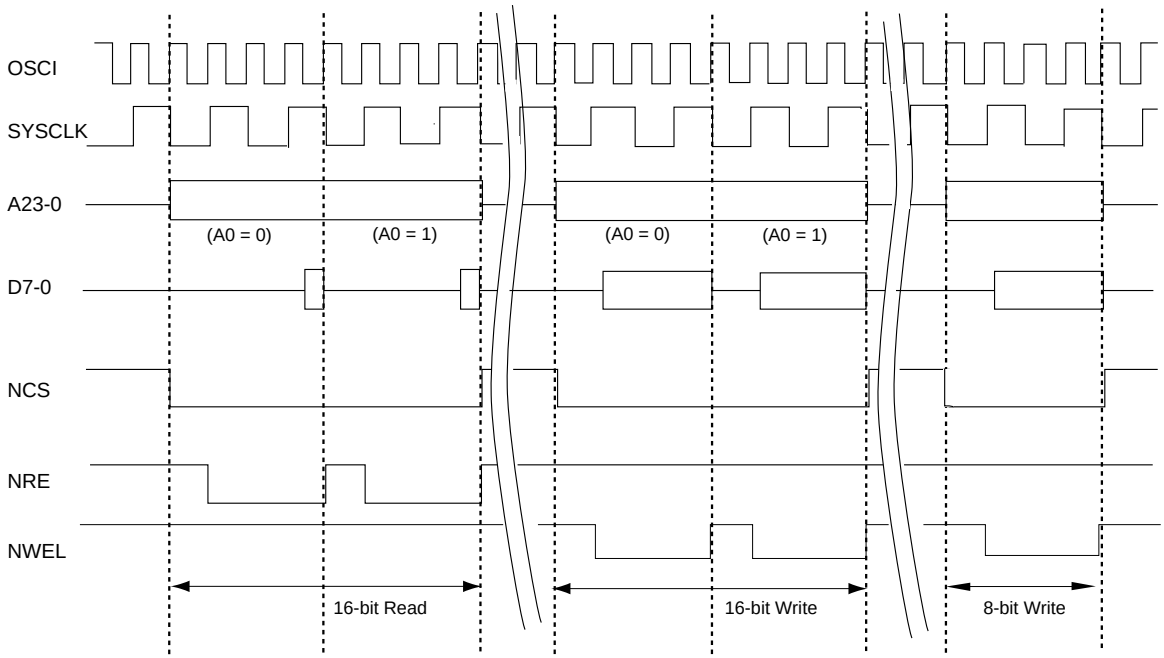


Figure 3-3-12 8-bit Bus Access Timing (1 Wait)

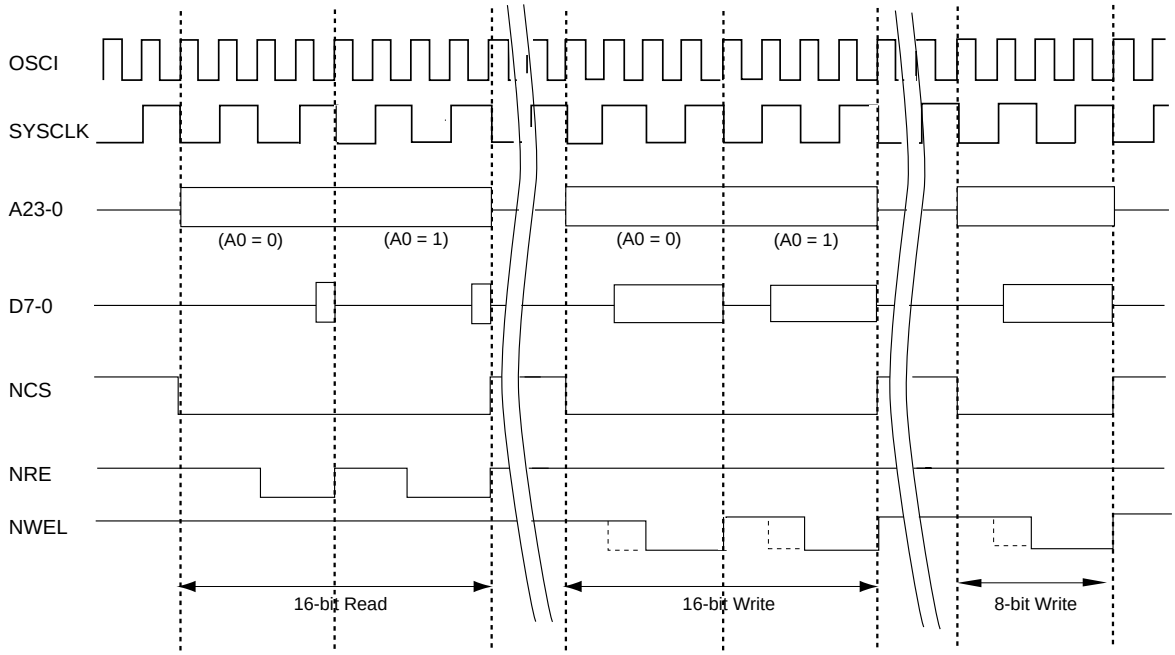


Figure 3-3-13 8-bit Bus Access Timing (1 Wait) (Late Access Mode)

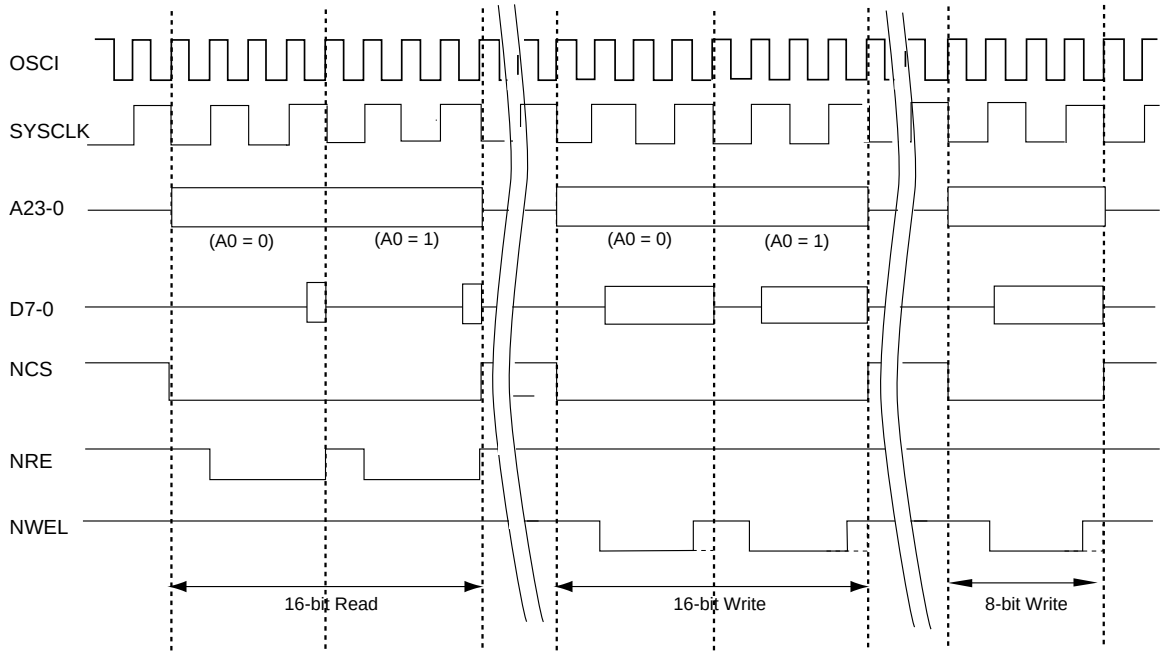


Figure 3-3-14 8-bit Bus Access Timing (1 Wait) (WE Short Mode)

■ Handshake Access Timing Using 16-bit Bus Mode (MnCBUS of MEMMDn =0)

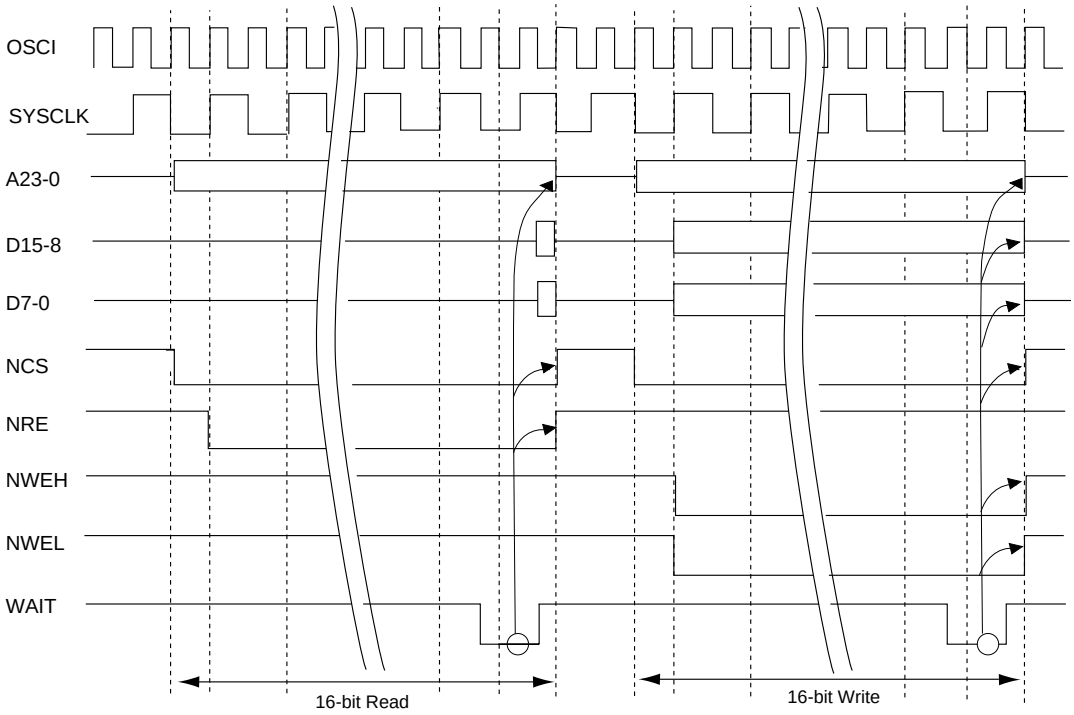


Figure 3-3-15 16-bit Bus Handshake Access Timing

■ Handshake Access Timing Using 8-bit Bus Mode (MnCBUS of MEMMDn =0)

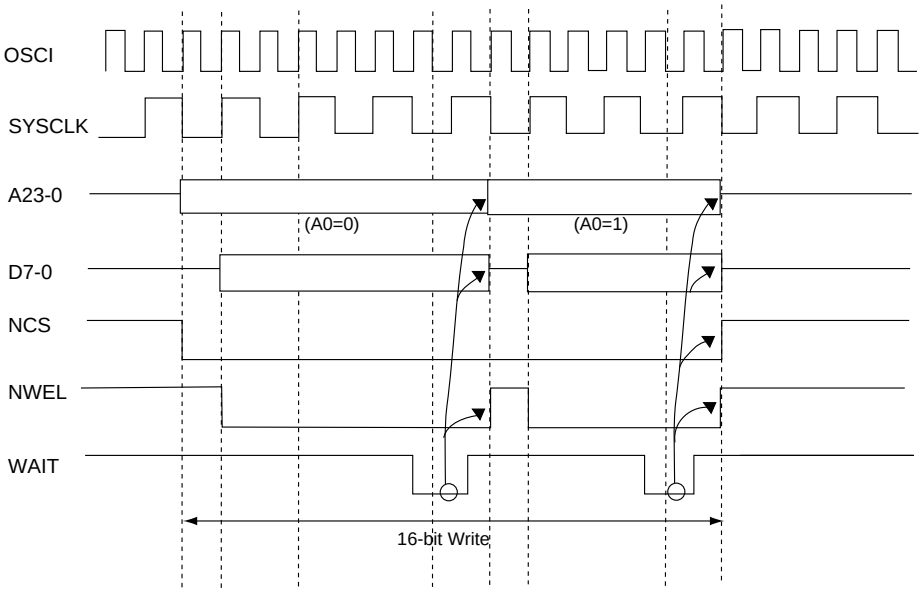


Figure 3-3-16 8-bit Bus Handshake Access Timing

■ Handshake Mode When Half Wait is Valid

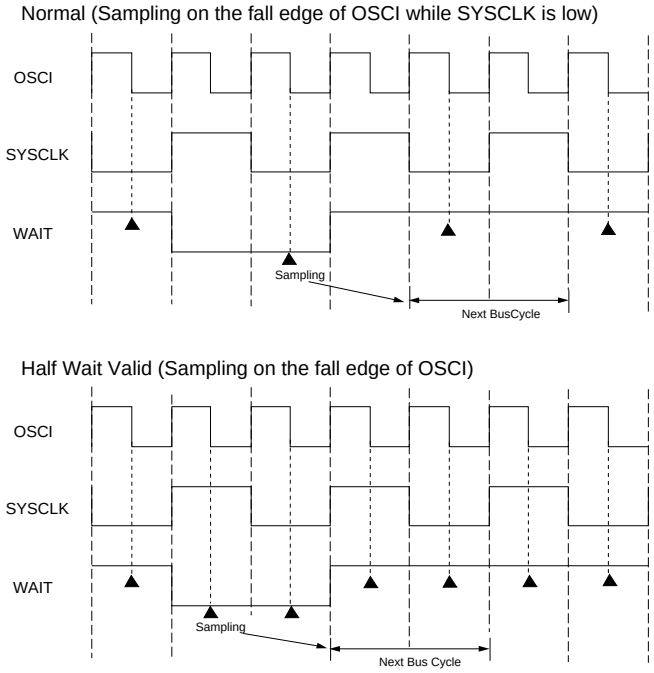


Figure 3-3-17 Handshake Access Timing at Half Wait

■ Wait Mask Function

The control that makes invalid a wait signal of the arbitrary cycle when a bus cycle starts, exists in [13:12] of each MEMMD register. It is not to acknowledge the LOW level of a wait signal of the previous cycle. It makes invalid 1 cycle by 01, 2 cycles by 10, and 3 cycles by 11.

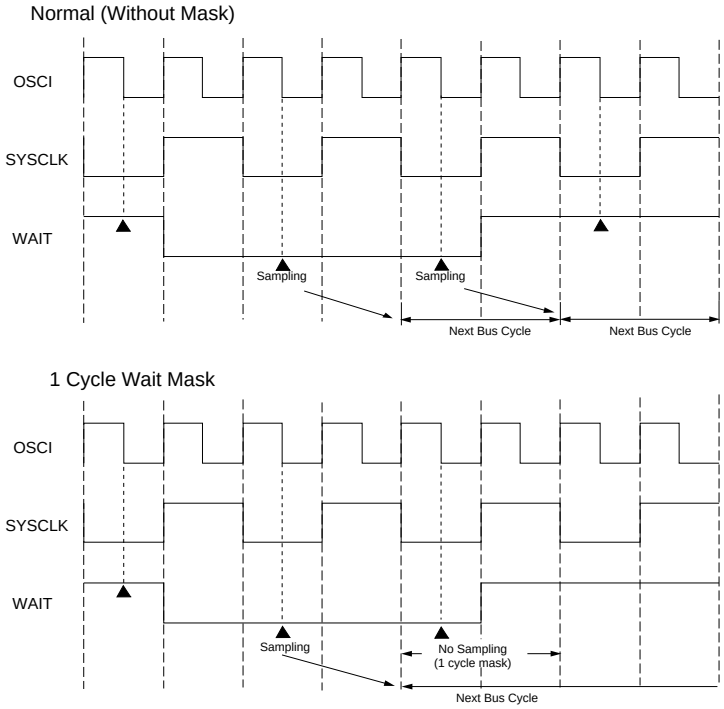


Figure 3-3-18 Wait Mask Function

3-4 DMA Support Function

3-4-1 Bus Arbitration

The MN102H74G/74F/74D/F74G has a bus arbitration function to exchange bus authority with external devices. Two external signals are used for bus arbitration; a bus master request signal (NBREQ) and a bus master enable signal (NBRACK). The external device accesses the memory after the MN102H74G/74F/74D/F74G releases bus using bus arbitration and the external device acquires the bus authority.

■ Bus Master Request Signal (NBREQ)

NBREQ is an output signal from external device. The MN102H74G/74F/74D/F74G samples NBREQ on the fall edge of OSCI while SYSCLK is high. When NBREQ is low, the MN102H74G/74F/74D/F74G releases a bus authority to external device by pulling NBRACK to low after its bus cycle ends. When NBREQ is high, the MN102H74G/74F/74D/F74G recovers the bus authority from external device by pulling NBRACK to high on the next rise edge of OSCI.

■ Bus Master Enable Signal (NBRACK)

NBRACK is an input signal to external device. Pulling NBRACK to low means that the MN102H74G/74F/74D/F74G releases the bus to the external device. Bus must be used always after pulling NBRACK to low.

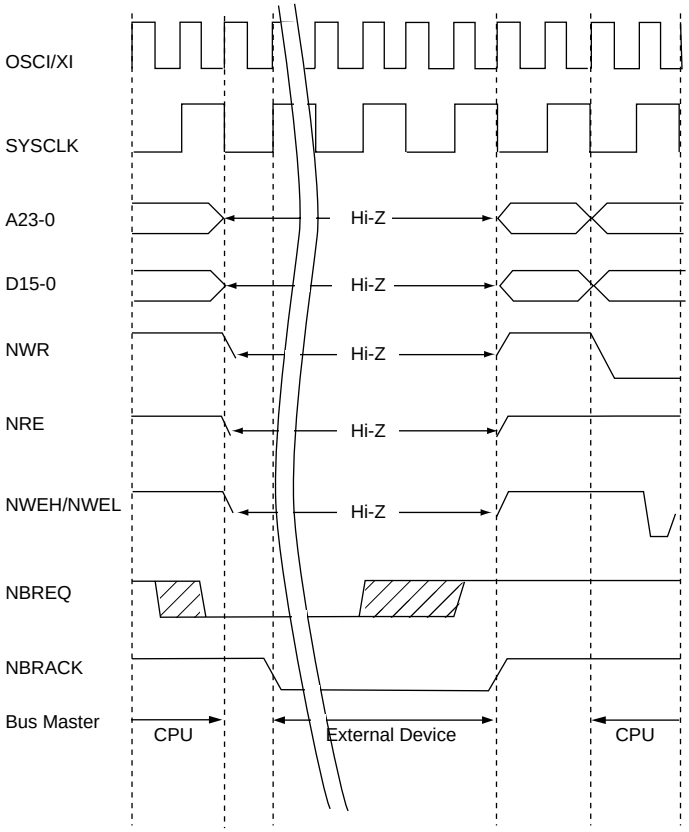


Figure 3-4-1 Bus Arbitration Timing

Chapter 4 Interrupts

4-1 Interrupt Controller

4-1-1 Overview

The interrupt controller consists of two interrupt methods; nonmaskable interrupt and maskable interrupt. The maskable interrupt has nine interrupt groups and eleven subgroups assigned from group 4 and group 9. One of seven levels can be set for each group. This provides hierarchical interrupt control.

Table 4-1-1 Interrupt Controller Overview

Group	Interrupt Vector (Number shows IDn bit position)			Control Register Register Abbreviation (Register Address)	
-		NRST Input		-	
Group 0	2	Undefined Instruction Interrupt		Nonmaskable Interrupt Control Register (Group 0) G0ICR (x'00FC40')	
	1	Watchdog Interrupt			
	0	NMI Interrupt			
Group 1	1	External Interrupt IRQ1		Maskable Interrupt Control Register (Group 1) G1ICR (x'00FC42')	
	0	External Interrupt IRQ0			
Group 2	1	External Interrupt IRQ3		Maskable Interrupt Control Register (Group 2) G2ICR (x'00FC44')	
	0	External Interrupt IRQ2			
Group 3	1	External Interrupt IRQ5		Maskable Interrupt Control Register (Group 3) G3ICR (x'00FC46')	
	0	External Interrupt IRQ4			
Group 4	1	Subgroup 4	2 A/D Conversion End Interrupt	Maskable Interrupt Control Register (Group 4) G4ICR (x'00FC48')	Subgroup Maskable Interrupt Control Register 40 SG40ICR (x'E00600')
			1 USB General-purpose Interrupt		
			0 USBSOF Interrupt		
	0	Reserved			-
Group 5	1	Subgroup 5	3 ATC3 Transfer End Interrupt	Maskable Interrupt Control Register (Group 5) G5ICR (x'00FC4A')	ATC Subgroup Interrupt Control Register ATIRQ (x'E00420')
			2 ATC2 Transfer End Interrupt		
			1 ATC1 Transfer End Interrupt		
			0 ATC0 Transfer End Interrupt		
			3 Serial 0 Transmission End Interrupt		Subgroup Maskable Interrupt Control Register 50 SG50ICR (x'E00602')
			2 Serial 0 Reception End Interrupt		
			1 Timer 3 Underflow		
			0 Timer 2 Underflow		
	0	Reserved			-
Group 6	1	Subgroup 6	3 Serial 1 Transmission End Interrupt	Maskable Interrupt Control Register (Group 6) G6ICR (x'00FC4C')	Subgroup Maskable Interrupt Control Register 60 SG60ICR (x'E00604')
			2 Serial 1 Reception End Interrupt		
			1 Timer 5 Underflow		
			0 Timer 4 Underflow		
			3 Timer 9 Underflow		Subgroup Maskable Interrupt Control Register 61 SG61ICR (x'E00606')
			2 Timer 8 Underflow		
			1 Timer 7 Underflow		
			0 Timer 6 Underflow		
	0	Reserved			-
Group 7	1	Subgroup 7	3 Timer 10 Compare/Capture B	Maskable Interrupt Control Register (Group 7) G7ICR (x'00FC4E')	Subgroup Maskable Interrupt Control Register 70 SG70ICR (x'E00608')
			2 Timer 10 Compare/Capture A		
			1 Timer 10 Underflow/Overflow		
			0 Serial 2 Reception End Interrupt		
			3 Timer 11 Compare/Capture B		Subgroup Maskable Interrupt Control Register 71 SG71ICR (x'E0060A')
			2 Timer 11 Compare/Capture A		
			1 Timer 11 Underflow/Overflow		
			0 Serial 2 Transmission End Interrupt		
	0	Reserved			-
Group 8	1	Timer 0 Underflow		Maskable Interrupt Control Register (Group 8) G8ICR (x'00FC50')	-
	0	Subgroup 8	3 Timer 12 Compare/Capture B		Subgroup Maskable Interrupt Control Register 80 SG80ICR (x'E0060C')
			2 Timer 12 Compare/Capture A		
			1 Timer 12 Underflow/Overflow		
			0 Serial 3 Transmission End Interrupt		
			3 USB Endpoint 4 Interrupt		Subgroup Maskable Interrupt Control Register 81 SG81ICR (x'E0061C')
			2 USB Endpoint 3 Interrupt		
			1 USB Endpoint 2 Interrupt		
			0 USB Endpoint 1 Interrupt		
Group 9	1	Timer 1 Underflow		Maskable Interrupt Control Register (Group 9) G9ICR (x'00FC52')	-
	0	Subgroup 9	3 Timer 13 Compare/Capture B		Subgroup Maskable Interrupt Control Register 90 SG90ICR (x'E0060E')
			2 Timer 13 Compare/Capture A		
			1 Timer 13 Underflow/Overflow		
			0 Serial 3 Reception End Interrupt		
			3 USB Endpoint 8 Interrupt		Subgroup Maskable Interrupt Control Register 91 SG91ICR (x'E0061E')
			2 USB Endpoint 7 Interrupt		
			1 USB Endpoint 6 Interrupt		
			0 USB Endpoint 5 Interrupt		

■ Cautions When Using Watchdog Interrupts

A watchdog interrupt detects an inadvertent program loop. It cannot be guaranteed that the CPU returns to the original operation with the RTI instruction after interrupt service routine. Do not return from the watchdog interrupt service routine. A watchdog interrupt occurs under the following conditions.

1. When the program does not execute with normal algorithm due to infinite loops or errors
2. When the CPU hangs up due to device errors or system errors (for example, the CPU hangs up because it cannot recognize the response signal while accessing the external device)

Under the second condition, the CPU enters the interrupt service routine without completing the instruction execution to end a bus cycle forcibly. In addition, the LSI cannot transfer the right data during ATC operation. Because of this, the program's normal operation cannot be guaranteed even though the program is returned from interrupt service routine.

■ Hardware Configuration

The interrupt controller contains the interrupt controller for peripherals and consists of ten interrupt control groups from Group 0 to Group 9. Group 0 is reserved for a nonmaskable interrupt.

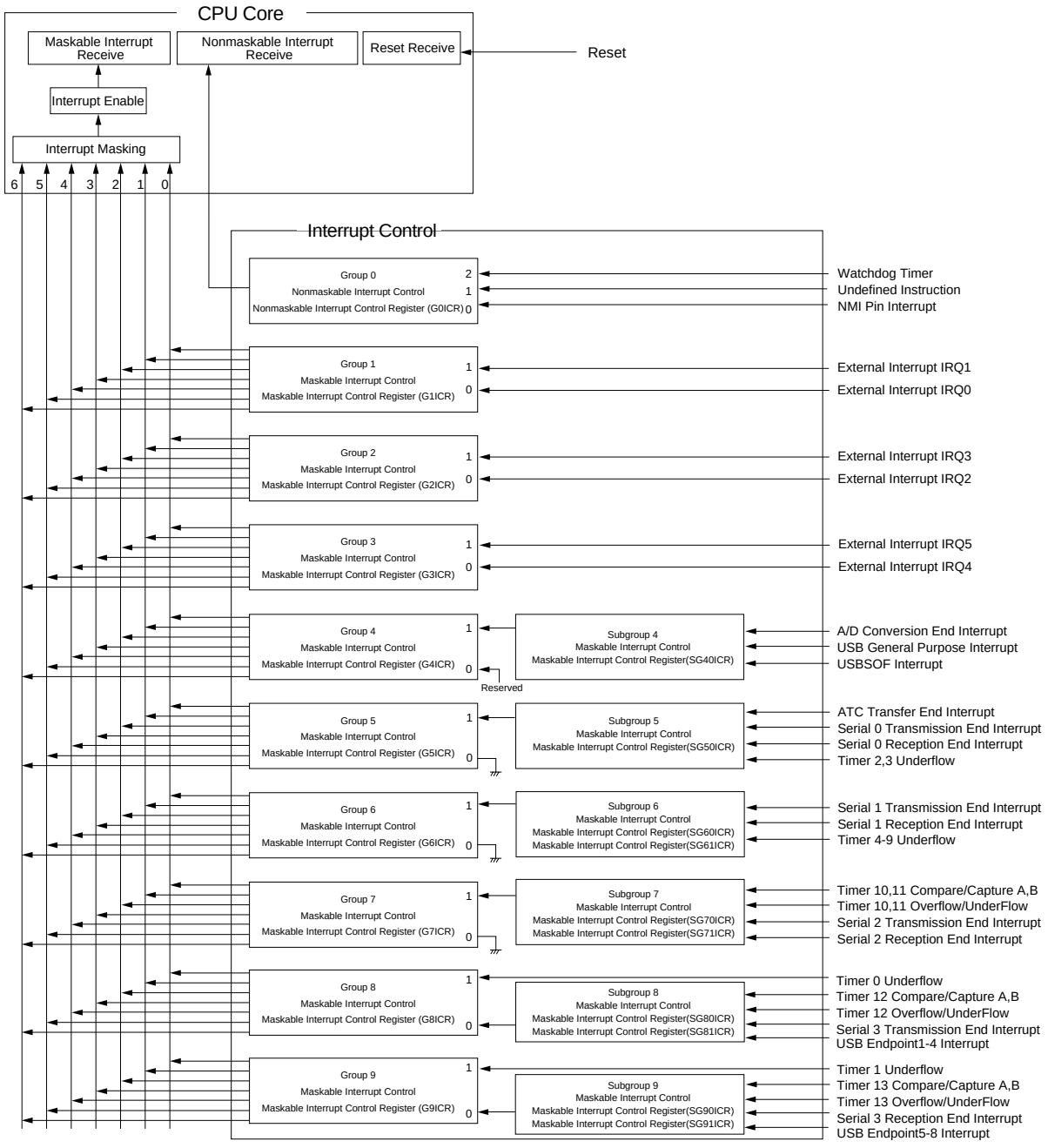


Figure 4-1-1 Interrupt Control Configuration

■ Interrupt Group and Mask Level

Each interrupt group has up to four interrupt vectors. Group 4 to group 9 can extend up to eight interrupt vectors by subgrouping. The interrupt level is set for each group except Group 0 (reserved for nonmaskable interrupt) using user program. The interrupt level is classified into seven levels in total. Group 1 has a priority if groups have the same interrupt level. (For example, if Group 6 and Group 7 have the level 2 and generate interrupt requests at the same time, the CPU accepts the interrupt of Group 6.) In addition, the interrupt handler determines the priority for each interrupt vector.

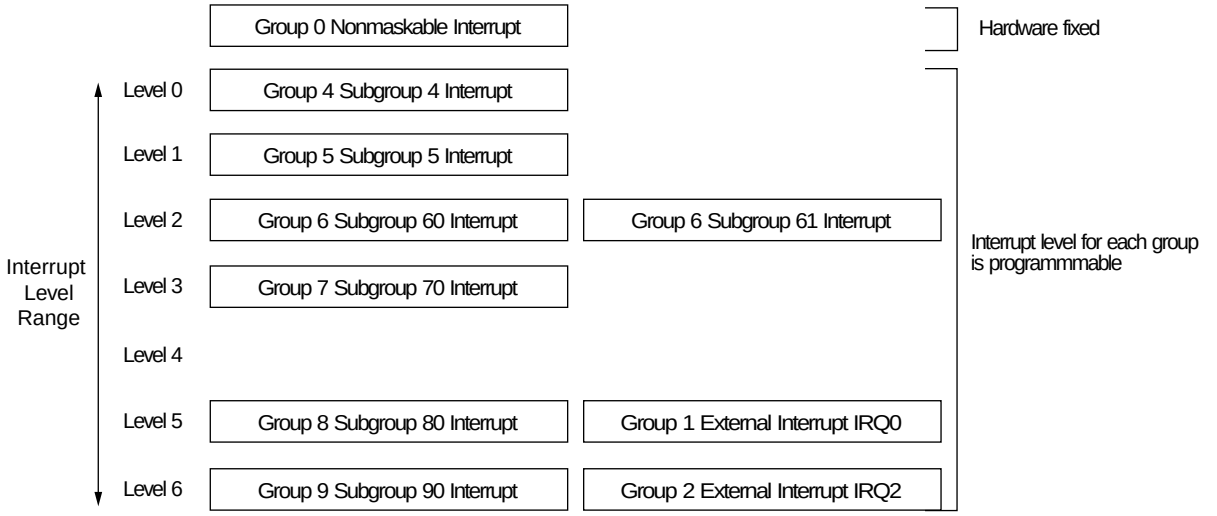


Figure 4-1-2 Interrupt Group

> Each Group except Group 0 has the maskable interrupt control register (GnICR) which controls an interrupt for each group. In addition, Group 4 to Group 9 have subgroup interrupt control registers (SGnmICR) which control an interrupt for each subgroup.

■ Interrupt Acceptance

The interrupt level corresponded for the generated interrupt vector is input to the CPU. The interrupt level flags [ILV2:0] of the maskable interrupt control register (GnICR) determine the interrupt level. The CPU accepts the interrupt when the interrupt level has a higher priority than the mask level which determined by IM[2:0] flags of PSW and the interrupt is enabled by IE of PSW. As a result of interrupt acceptance, the interrupt level is updated to the level set by IM[2:0] flags of PSW and the interrupt is disabled by IE flag of PSW. The CPU , however, always accepts a reset input or a nonmaskable interrupt regardless of the mask level or IE flag status.

■ Interrupt Service Routine Sequence

The whole interrupt service routine is the sequence of interrupt request, interrupt acceptance, hardware process and interrupt service routine. After the interrupt acceptance, the program counter and processor status word are pushed to the stack area in the hardware process, and the program is branched to the interrupt preprocess program written in x'080008'. The interrupt preprocess analyzes which interrupt is accepted, read the first address of the accepted interrupt program from the entry address table of each interrupt service routine created in the program, and then branches to that first address. After interrupt service routine ends, the interrupt postprocess returns the register value pushed at interrupt acceptance and the hardware process returns the program counter and processor status word.

> ! In the MN102H74G/74F/74D/F74G, the sequence of maskable interrupt and the sequence of nonmaskable interrupt are the same.

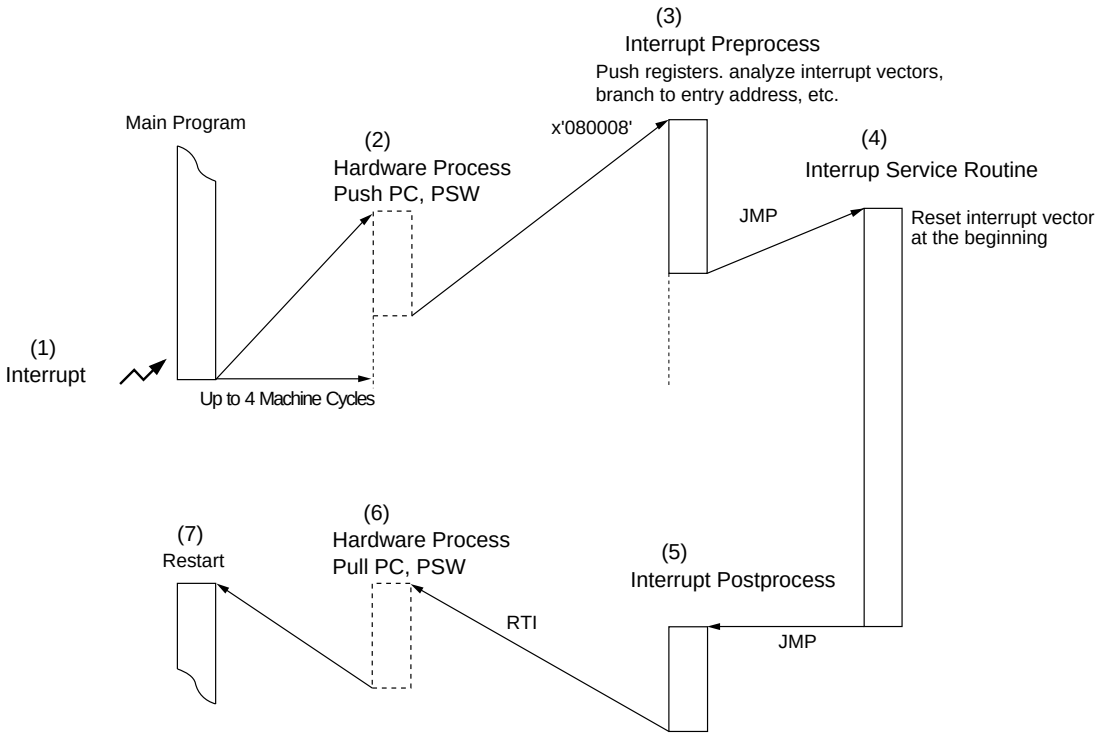


Figure 4-1-3 Interrupt Service Routine Sequence

4-1-2 Control Registers

These registers control interrupts; maskable interrupt control registers (GnICR), a nonmaskable interrupt control register (G0ICR), an interrupt accept group number register (IAGR) and sub-group maskable control registers (SGnmICR).

■ Maskable Interrupt Control Registers (GnICR x'00FC42' to x'00FC52')

The maskable interrupt control register (GnICR) is allocated in each group except Group 0 and controls an interrupt input to its group. The GnICR register consists of interrupt level flags (LVn), interrupt enable flags (IEn), interrupt request flags (IRn) and interrupt detect flags (IDn). For example, TM0IR of the G8ICR register becomes 1 when timer 0 underflows. Then, an interrupt request is output to the CPU if TM0IE is 1. The CPU accepts the interrupt request when the interrupt level (set by IRQ8LV[2:0] of G8ICR) is lower than the mask level (set by IM[2:0] of PSW) and the interrupt enable flag (IE) of PSW is 1.

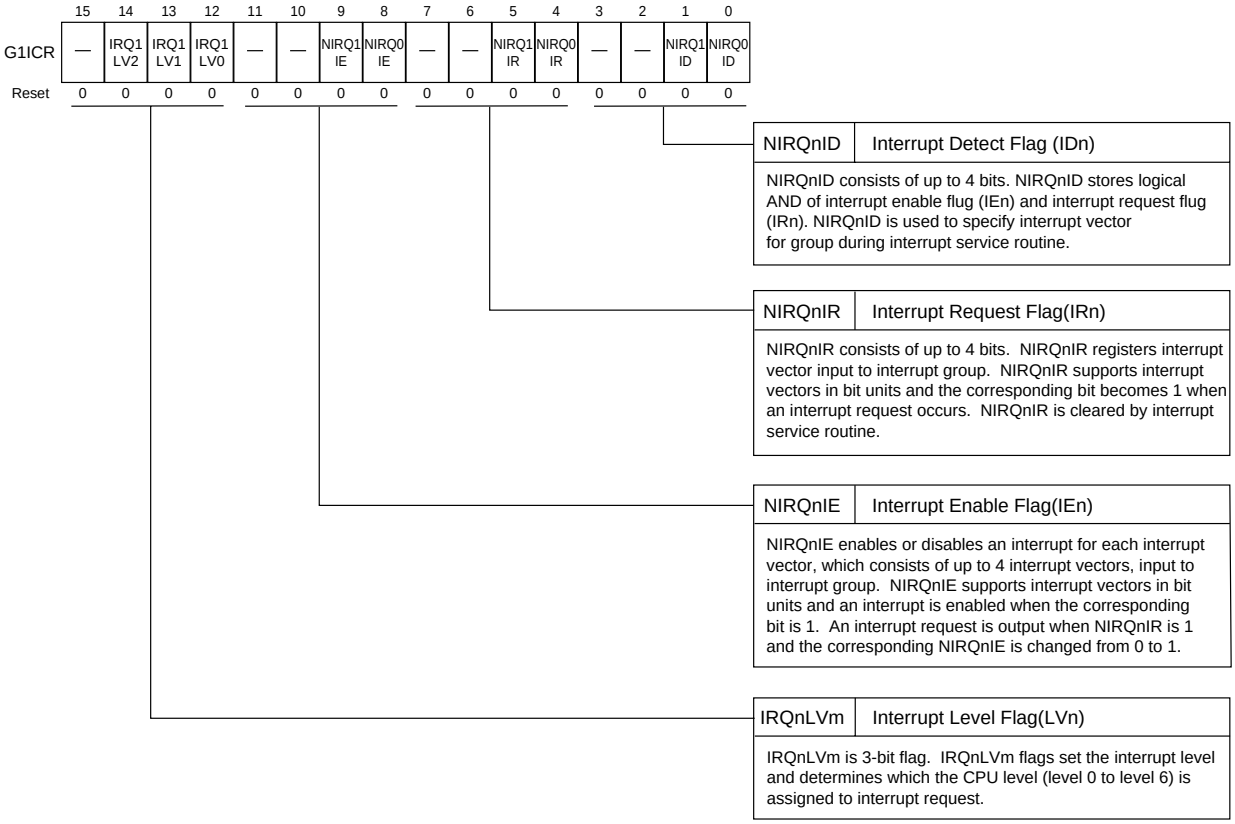

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G1ICR — IRQ1 LV2 IRQ1 LV1 IRQ1 LV0 — — NIRQ1 IE NIRQ0 IE — — NIRQ1 IR NIRQ0 IR — — NIRQ1 ID NIRQ0 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NIRQnID Interrupt Detect Flag (IDn)
NIRQnID consists of up to 4 bits. NIRQnID stores logical AND of interrupt enable flug (IEn) and interrupt request flug (IRn). NIRQnID is used to specify interrupt vector for group during interrupt service routine.
NIRQnIR Interrupt Request Flag(IRn)
NIRQnIR consists of up to 4 bits. NIRQnIR registers interrupt vector input to interrupt group. NIRQnIR supports interrupt vectors in bit units and the corresponding bit becomes 1 when an interrupt request occurs. NIRQnIR is cleared by interrupt service routine.
NIRQnIE Interrupt Enable Flag(IEn)
NIRQnIE enables or disables an interrupt for each interrupt vector, which consists of up to 4 interrupt vectors, input to interrupt group. NIRQnIE supports interrupt vectors in bit units and an interrupt is enabled when the corresponding bit is 1. An interrupt request is output when NIRQnIR is 1 and the corresponding NIRQnIE is changed from 0 to 1.
IRQnLVm Interrupt Level Flag(LVn)
IRQnLVm is 3-bit flag. IRQnLVm flags set the interrupt level and determines which the CPU level (level 0 to level 6) is assigned to interrupt request.

Figure 4-1-4 Maskable Interrupt Control Register (GnICR)

When an interrupt is urgent or its process time is short, select the high interrupt level (The values of the interrupt level flags (LVn) are small.) The CPU does not accept the interrupt of group where the LVn flags are set to 7.

The IE flag of PSW must be 0 when writing the GnICR register and the G0ICR register. Refer to “4-1-6 Interrupt Acceptance” on page 4-18.

Clear the NIRQnIR bit of maskable interrupt control register (GnICR) after clearing IR bit of subgroup maskable interrupt control register (SGnICR) for the group which has subgroup (4, 5, 6, 7, 8, 9). Interrupt request flag is not cleared other than this procedure.

Table 4-1-2 Mask Level and Interrupt Level

Interrupt Mask Level (PSW)	Received Interrupt Level	Interrupt Level (ICRn)	Priority
IMn		ILVn	
0	Nonmaskable Interrupt (NMI) only	Reserved	High
1	Level 0, NMI	0	↑
2	Level 0-1, NMI	0-1	
3	Level 0-2, NMI	0-2	
4	Level 0-3, NMI	0-3	
5	Level 0-4, NMI	0-4	
6	Level 0-5, NMI	0-5	
7	Level 0-6, NMI	0-6	Low

When an interrupt does not exist even though the CPU accepts the interrupt request, the IAGR register is 0. In this case, UNIF, WDIF and NMIF remain 0 so that the interrupt is check using software in the interrupt handler if the process is required. The interrupt may not exist if the IR flags are cleared without the interrupt service routine.

■ Nonmaskable Interrupt Control Register (G0ICR x'00FC40')

The nonmaskable interrupt control register (G0ICR) is allocated in Group 0 and stores a nonmaskable interrupt.

	15							8	7							0
G0ICR	0	0	0	0	0	0	0	0	0	0	0	0	0	UNIF	WDIF	NMIF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-1-5 Nonmaskable Interrupt Control Register (G0ICR)

External Nonmaskable Interrupt Request Flag (UNIF)

The NMIF flag becomes 1 if the negative edge (pulse width with more than 4 cycles of oscillator) is input to the external NNMI pin.

Watchdog Timer Overflow Interrupt Request Flag (WDIF)

The WDIF flag becomes 1 when the watchdog timer overflows.

Undefined Instruction Interrupt Request Flag (UNIF)

The UNIF flag becomes 1 when the undefined instruction is executed.

Normally, the watchdog timer is used to detect program errors. Therefore, allocate the program of clearing timer within 2^{16}- cycle internal to other areas except the interrupt service routine. The CPU mode control register (CPUM) controls the watchdog timer.

The CPU cannot return to the original operation after interrupt process because the watchdog timer overflow interrupt is used for error detection. The re-execution from initialization or error process program must be programmed as the watchdog timer overflow interrupt process. Use a timer interrupt when returning from interrupt. Refer to "2-6 Error Detection Function".

■ Interrupt Acceptance Group Number Register (IAGR x'00FC0E')

The interrupt acceptance group number register (IAGR) is a 16-bit register. The IAGR register stores the group number to specify which an interrupt occurs from which group when the interrupt is accepted.

	15							8	7							0
IAGR	0	0	0	0	0	0	0	0	0	GN4	GN3	GN2	GN1	GN0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-1-6 Interrupt Acceptance Group Number Register (IAGR)

■ Subgroup Maskable Interrupt Control Registers (SGnmICR x'E00600' to x'E00610')

The subgroup maskable interrupt control register (SGnmICR) is allocated in each group of Group 4 to Group 9 to increase the number of interrupt vectors and controls an interrupt input to its subgroup. The SGnmICR register consists of interrupt enable flags (IEn), interrupt request flags (IRn) and interrupt detect flags (IDn). For example, SC0RIR of the SG50ICR register becomes 1 when a serial reception end interrupt occurs. An interrupt request is output to Group 5 and NIRQ9IR of the G5ICR register becomes 1 when SC0RIE is set to 1. Then the interrupt request is input to the CPU when NIRQ9IE of the G5ICR register is 1.

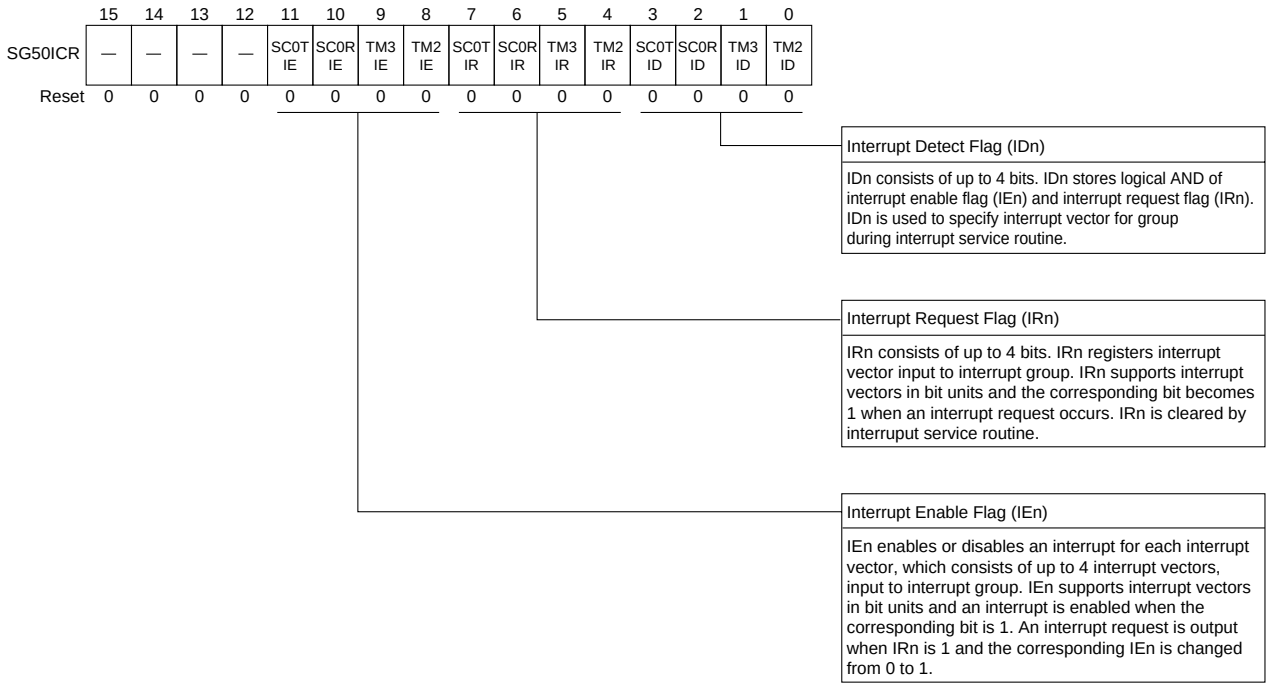

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG50ICR — — — — SC0TIE SC0RIE TM3IE TM2IE SC0TIR SC0RIR TM3IR TM2IR SC0TID SC0RID TM3ID TM2ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Interrupt Detect Flag (IDn)
IDn consists of up to 4 bits. IDn stores logical AND of interrupt enable flag (IEn) and interrupt request flag (IRn). IDn is used to specify interrupt vector for group during interrupt service routine.
Interrupt Request Flag (IRn)
IRn consists of up to 4 bits. IRn registers interrupt vector input to interrupt group. IRn supports interrupt vectors in bit units and the corresponding bit becomes 1 when an interrupt request occurs. IRn is cleared by interruput service routine.
Interrupt Enable Flag (IEn)
IEn enables or disables an interrupt for each interrupt vector, which consists of up to 4 interrupt vectors, input to interrupt group. IEn supports interrupt vectors in bit units and an interrupt is enabled when the corresponding bit is 1. An interrupt request is output when IRn is 1 and the corresponding IEn is changed from 0 to 1.

Figure 4-1-7 Subgroup Maskable Interrupt Control Register (SG50ICR)

4-1-3 Interrupt Level

Interrupt level is set for each group. It is determined whether a maskable interrupt is accepted or not depending on the interrupt enable flag (IE) setup of PSW and the mask level (IM[2:0]).

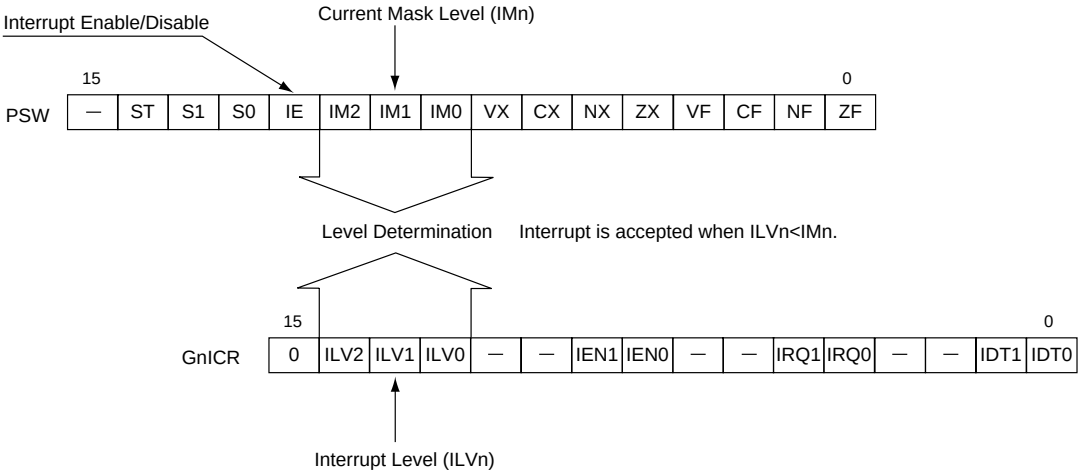


Figure 4-1-8 Interrupt Acceptance Determination

The following describes the sequence of occurring an interrupt request and accepting it.

(1) The corresponding interrupt request flag (IRQ[1:0]) of the maskable interrupt control register (GnICR) becomes 1 when an interrupt request occurs.

(2) The interrupt request is output to the CPU when the interrupt enable flag (IEN[1:0]) is 1.

(3) The interrupt level set by the interrupt level flag (ILV[2:0]) is output to the CPU.

(4) The interrupt is accepted when the level of the interrupt request is higher than the mask level (IM[2:0]) of PSW and the IE flag of PSW is 1.

To clear interrupt request in the interrupt service routine, write 0 to the corresponding bit. In this case, the contents of ILVn, IENn and untreated interrupt request flag (IRQn) must be maintained. Therefore, clear the corresponding bit of IRQn in ICRn which is read to specify the interrupt vector in the group, and write to ICRn. The interrupt request which is generated during this operation is maintained after writing to ICRn.

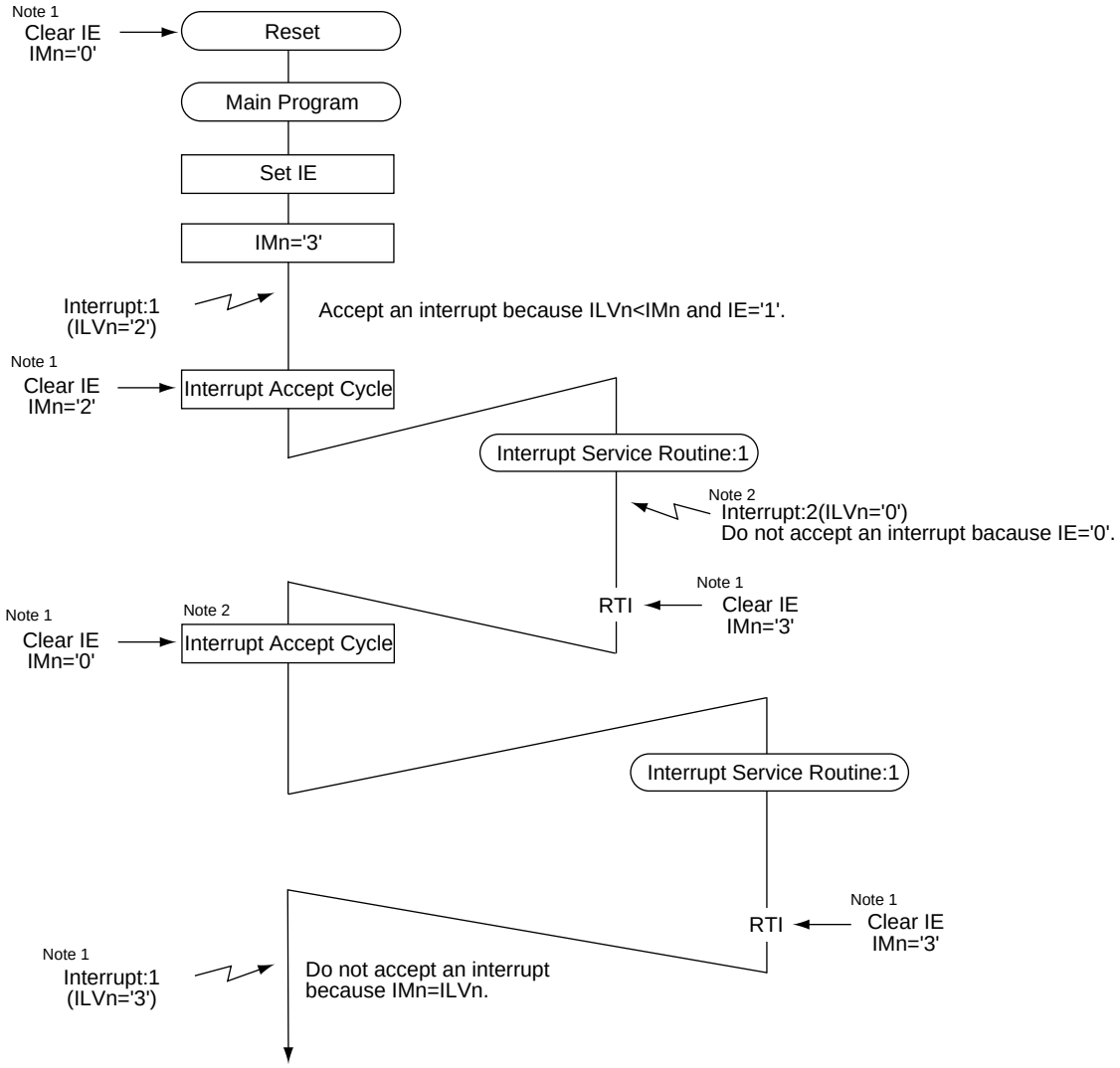
The maskable interrupt acceptance is disabled automatically, except the interrupt enable flag (IE) of PSW is reset to 0 when the interrupt is accepted and multiple interrupt is enabled.
Refer to “4-1-11 Multiple Interrupt”.

The IE flag becomes 0 and an interrupt is disabled under one of the following conditions.
When 0 is written to IE of PSW by program
When a maskable interrupt is accepted when IE is 1
When a nonmaskable interrupt is accepted
When a reset is input
When a reset is input and IE of PSW is 0

The IE flag becomes 1 and an interrupt is enabled under one of the following conditions.
When 1 is written to IE of PSW by program
When a maskable interrupt is accepted, the RTI instruction is executed at the end of its interrupt service routine and IE of PSW is 1

The mask level is changed under one of the following conditions.
Write the new value to the IMn flag of PSW by program
Accept a reset input or a nonmaskable interrupt (IMn = x’000’)
When a maskable interrupt is accepted, the mask level is the same as that maskable interrupt level. The mask level becomes the level before accepting an interrupt when RTI instruction is executed at the end of interrupt service routine.

When a nonmaskable interrupt and a maskable interrupt occur at the same time, a nonmaskable interrupt has a priority. The mask level after an interrupt occurs, however, is changed to the maskable interrupt level.



Note 1: The process occurs in hardware.
Note 2: Do not accept an interrupt occurred before ending the interrupt service routine execution because IE is cleared.

Figure 4-1-9 Maskable Interrupt Process Sequence (without Multiple Interrupts)

4-1-4 External Interrupt

External interrupts are controlled by Group 1 to Group 9.

The group interrupt edge specification (EXTMD) sets the interrupt trigger conditions. The EXTMD register sets the edge/level of the input signal for each pin. (The initial values are '00': low level.)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTMD0	NIRQ7 TG1	NIRQ7 TG0	NIRQ6 TG1	NIRQ6 TG0	NIRQ5 TG1	NIRQ5 TG0	NIRQ4 TG1	NIRQ4 TG0	NIRQ3 TG1	NIRQ3 TG0	NIRQ2 TG1	NIRQ2 TG0	NIRQ1 TG1	NIRQ1 TG0	NIRQ0 TG1	NIRQ0 TG0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NIRQ0TG1	NIRQ0TG0	NIRQ0 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ1TG1	NIRQ1TG0	NIRQ1 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ2TG1	NIRQ2TG0	NIRQ2 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ3TG1	NIRQ3TG0	NIRQ3 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ4TG1	NIRQ4TG0	NIRQ4 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ5TG1	NIRQ5TG0	NIRQ5 Interrupt Trigger Condition
0	0	Low level
0	1	High level
1	0	Negative (Fall) edge
1	1	Positive (Rise) edge

NIRQ6TG1	NIRQ6TG0	Reserved

NIRQ7TG1	NIRQ7TG0	Subgroup 4 Interrupt Trigger Condition
0	0	Low level
0	1	Reserved
1	0	Reserved
1	1	Reserved

Figure 4-1-10 External Interrupt Edge Specification 0 (EXTMD0: x'00FC56', R/W)

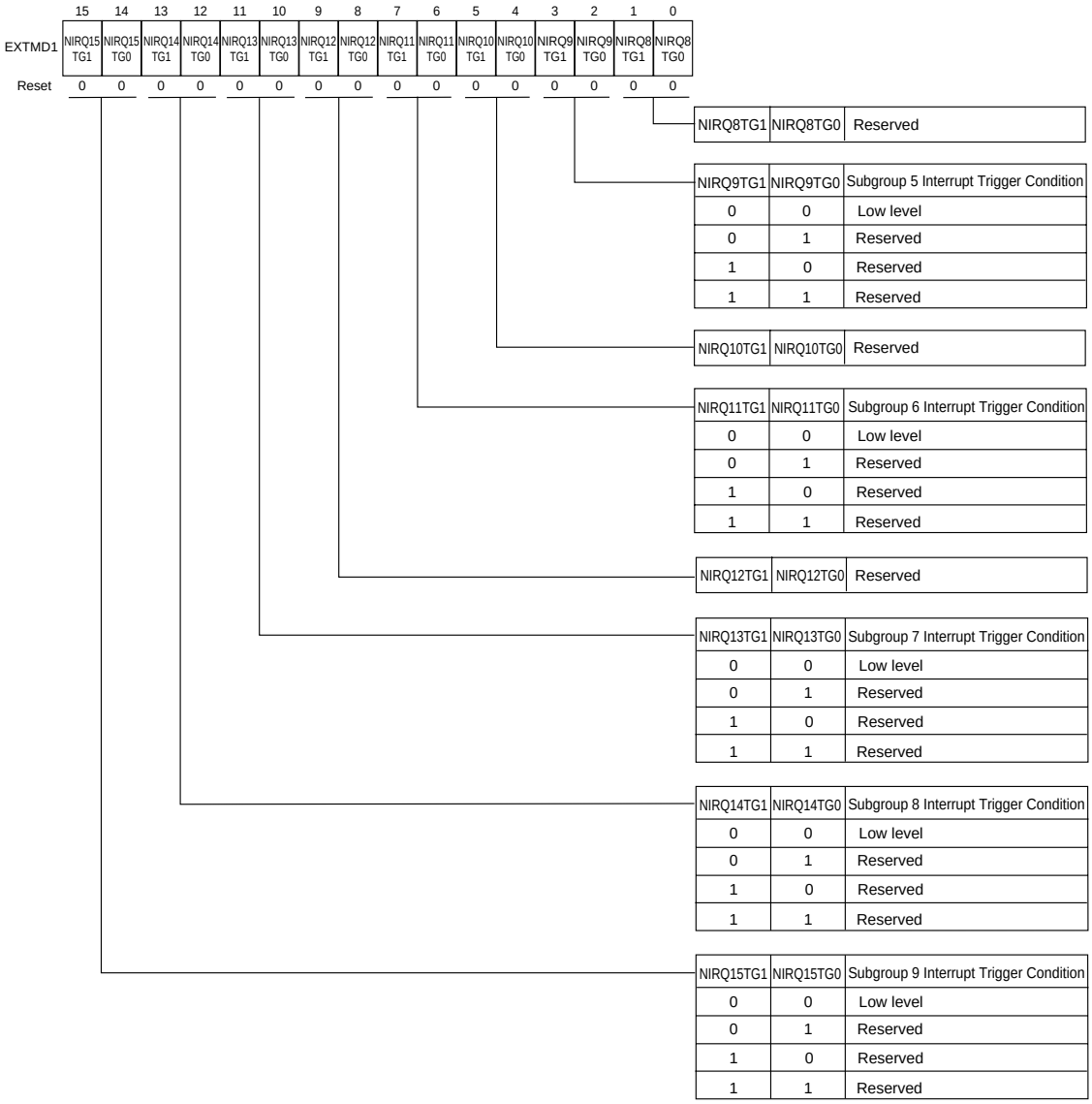

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
EXTMD1 NIRQ15 TG1 NIRQ15 TG0 NIRQ14 TG1 NIRQ14 TG0 NIRQ13 TG1 NIRQ13 TG0 NIRQ12 TG1 NIRQ12 TG0 NIRQ11 TG1 NIRQ11 TG0 NIRQ10 TG1 NIRQ10 TG0 NIRQ9 TG1 NIRQ9 TG0 NIRQ8 TG1 NIRQ8 TG0
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
NIRQ8TG1 NIRQ8TG0 Reserved
NIRQ9TG1 NIRQ9TG0 Subgroup 5 Interrupt Trigger Condition
0 0 Low level
0 1 Reserved
1 0 Reserved
1 1 Reserved
NIRQ10TG1 NIRQ10TG0 Reserved
NIRQ11TG1 NIRQ11TG0 Subgroup 6 Interrupt Trigger Condition
0 0 Low level
0 1 Reserved
1 0 Reserved
1 1 Reserved
NIRQ12TG1 NIRQ12TG0 Reserved
NIRQ13TG1 NIRQ13TG0 Subgroup 7 Interrupt Trigger Condition
0 0 Low level
0 1 Reserved
1 0 Reserved
1 1 Reserved
NIRQ14TG1 NIRQ14TG0 Subgroup 8 Interrupt Trigger Condition
0 0 Low level
0 1 Reserved
1 0 Reserved
1 1 Reserved
NIRQ15TG1 NIRQ15TG0 Subgroup 9 Interrupt Trigger Condition
0 0 Low level
0 1 Reserved
1 0 Reserved
1 1 Reserved

Figure 4-1-11 External Interrupt Edge Specification 1 (EXTMD1: x'00FC58', R/W)

4-1-5 NNMI Pin Interrupt

The MN102H74G/74F/74D/F74G supports a NNMI interrupt. A NNMI interrupt occurs on the negative edge of NNMI pin.

A NNMI interrupt does not occur when bus is released or handshake access wait mode is selected. However bus is also released during ATC transfer, it is possible to suspend transfer by NNMI interrupt.
Refer to “9-3-11 Transfer Suspend Using Nonmaskable Interrupt (NMI)”.

4-1-6 Interrupt Acceptance

The CPU executes hardware process automatically when an interrupt is accepted. After that, the CPU starts interrupt service routine by software interrupt handler.

(1) Push Program return address and processor status word to the stack area when the CPU accepts an interrupt. Hardware executes this process automatically.

> Select the even address for stack pointer because the word transfer from the odd address is not allowed.

> Regardless of a maskable interrupt or a nonmaskable interrupt, 3 words of 24-bit PC and 16-bit PSW are pushed.

Push the contents of the program counter (return address) to the stack (SP-4).
Push the contents of the processor status word (PSW) to the stack (SP-6).
Update the PSW data. Copy the interrupt level (ILVn) of the accepted interrupt group to IMn and clear the IE flag of PSW.
Update the stack pointer data. (SP-6 to SP)
Update the PC data. (PC = x'080008')

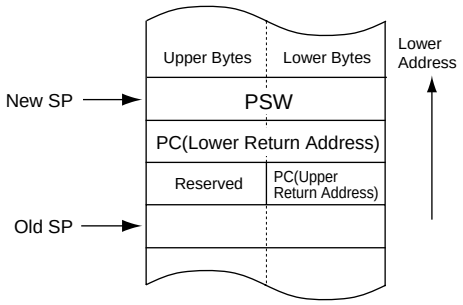


Figure 4-1-12 Stack Pointer (SP) Operation at Interrupt

> Assign the address register (A3) to the stack pointer. Assign the interrupt program on x'080008'.

(2) Assign the interrupt preprocess program on x'080008'. The user program executes this process.

Push the data of registers used for the interrupt service routine to the stack pointer.
Read the interrupt accept group number register (IAGR) and the maskable interrupt control register (GnICR) or the subgroup maskable interrupt control register (SGnmICR), and then specify the interrupt vector to be accepted.
Read the first address of the interrupt service routine from the entry address table created in the program.
Branch to the address to be read.

The following shows the example of the interrupt preprocess program. Programming differs depending on the real user programs. The interrupt preprocess program of a normal interrupt and that of a nonmaskable interrupt are same. Refer to "MN102H Series C Compiler User's Manual" for the interrupt function table irq_vec_tbl.

```
        org     Reset +0x000008         Move the program to x'080008' when an interrupt occurs.
Interrupt:
        add     -14, a3
        mov     a0, (10, a3)
        movx    d0, (6, a3)
        movx    d1, (2, a3)
        mov     mdr, d0                 Interrupt Preprocess
        mov     d0, (a3)
        mov     (IAGR), d0
        mov     _irq_vec_tbl, A0
        mov     (d0, a0), a0
        jsr     (a0)
```

As the following example shows, an interrupt must be disabled before interrupt level flags and interrupt enable flags of the maskable interrupt control register (GnICR) are set.

```
        ......                  ;
        add     0xf7ff, psw     ; Clear IE of PSW
        nop                     ;       Insert nops to access GnICR completely while IE of PSW is cleared
        nop                     ;
        mov     d0, (GnICR)     ; Change LVn/IE
        mov     (GnICR), d0     ; Insert to change GnICR completely
        or      0x0800, psw     ; Set IE of PSW
        ......                  ;
```

Because IE is 0 during interrupt handler unless IE of PSW is set, it is not necessary that an interrupt is disabled by clearing IE. The nop instructions as shown above can be any instructions except instructions of changing IE of PSW or LVn/IE of GnICR. Two nop instruction are inserted to secure the minimum cycles for changing the IE flag of PSW. Therefore, two nop instructions or more can be inserted.

4-1-7 External Interrupt Setup Example

An external interrupt occurs on the negative (fall) edge of the external interrupt pin NIRQ0.

After reset, the external interrupt edge specification register (EXTMD) selects low level to generate an interrupt request and NIRQ0IR of the maskable interrupt control register (G1ICR) becomes 0.

■ Interrupt Enable Setup

(1) Set NIRQ0 interrupt trigger condition. Set NIRQ0TG[1:0] of the EXTMD0 register to '10' (negative edge).

EXTMD0: x'00FC56'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIRQ7 TG1	NIRQ7 TG0	NIRQ6 TG1	NIRQ6 TG0	NIRQ5 TG1	NIRQ5 TG0	NIRQ4 TG1	NIRQ4 TG0	NIRQ3 TG1	NIRQ3 TG0	NIRQ2 TG1	NIRQ2 TG0	NIRQ1 TG1	NIRQ1 TG0	NIRQ0 TG1	NIRQ0 TG0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

(2) Enable an interrupt after clearing all prior interrupt requests. Set IRQ1LV[2:0] flags of the G1ICR register to the interrupt level, NIRQ0IR to 0 and NIRQ0IE to 1. In this example, the interrupt level is 4.

G1ICR: x'00FC42'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IRQ1 LV2	IRQ1 LV1	IRQ1 LV0	-	-	NIRQ1 IE	NIRQ0 IE	-	-	NIRQ1 IR	NIRQ0 IR	-	-	NIRQ1 ID	NIRQ0 ID
-	1	0	0	-	-	0	1	-	-	0	0	-	-	0	0

(3) Enable an interrupt by setting the interrupt enable flag (IE) of PSW to 1 and the mask level (IMn) to 7 ('111').

```
        org     0x0000          ; Specify the start address
Start
        jmp     (init)          ; Branch to the initialization routine of register
        org     0x0050
init
        mov     0x00, d0
        mov     d0, d1
        mov     d0, d2
        mov     d0, d3
        mov     d0, a0
        mov     d0, a1
        mov     d0, a2
        mov     0xe800, a3      ; Set the stack pointer to 0xe800
irq0                            ; External interrupt service routine
        mov     0x0002, d2      ; Set the NIRQ0 interrupt trigger condition to negative edge
        mov     d2, (extmd0)    ;
        mov     0x4100, d1      ; Set the interrupt level of G1ICR to 4, NIRQ0IR to 0
        mov     d1, (g1icr)     ;           NIRQ0IE to 1
        mov     0x0f00, d0      ; Enable an interrupt by setting IMn of PSW to 7, IE of PSW to 1
        mov     d0, psw         ;
```

Thereafter, an interrupt occurs on the negative (fall) edge of the external interrupt pin NIRQ0. The program branches to x'080008' when an interrupt occurs and is accepted.

■ Interrupt Service Routine

(4) Specify the interrupt group by reading the interrupt accept group number register (IAGR) during interrupt preprocessing.

> Normally, the program generates the interrupt start address and branches to that address.

(5) Specify the interrupt vector by reading G1ICR. Check NIRQ0ID of G1ICR with the bit test instruction (BTST). Execute the interrupt service routine if NIRQ0ID is 1.

(6) Clear NIRQ0IR of G1ICR.

(7) Return to the original program with the RTI instruction after the interrupt service routine ends.

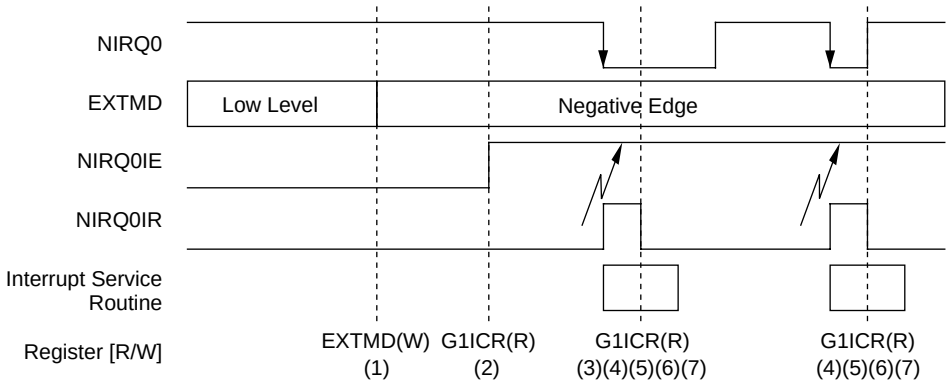


Figure 4-1-13 External Interrupt Setup Timing

> The IMn and IE of PSW become the interrupt level and 0 respectively and the CPU does not accept multiple interrupts during the interrupt service routine. The CPU accept a nonmaskable interrupt only unless PSW is set.

4-1-8 Watchdog Timer Interrupt Setup Example

An interrupt occurs by watchdog timer.

The watchdog timer start operation by setting the WDRST flag of the CPU mode control register (CPUM) to '0' (enable) after reset. It is necessary to clear the watchdog timer in the main program because a nonmaskable interrupt occurs when the watchdog timer overflows.

■ Interrupt Enable Setup

(1) Enable an interrupt by setting the IE flag and the IMn flags of PSW to '1' and '111' respectively.

(2) Start the watchdog timer by clearing the WDRST flag of the CPUM register.

CPUM: x'00FC00'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD RST	WD MD1	WD MD0	-	-	-	-	-	-	-	-	OSC ID	STOP	HALT	OSC1	OSC0
0	0	0	-	-	-	-	-	-	-	-	0	0	0	0	0

If WDMD1 and WDMD0 are '00', a watchdog timer occurs when the watchdog timer counts 2^{17} BOSC cycles (5.461 ms with 12-MHz external oscillator). The following is WDMD[1:0] setting.

00: 2^{17}
01: Reserved
10: 2^{13}
11: 2^{15}

■ Watchdog Timer Clear

(3) Clear the watchdog timer by setting the WDRST flag of the CPUM register to 1 and immediately clearing it to 0.

Normally, clear the watchdog timer before an interrupt occurs.

■ Interrupt Service Routine

The program branches to x'080008' when an interrupt is generated and accepted.

(4) Specify the interrupt group by reading the interrupt accept group number register (IAGR) during interrupt preprocessing.

> Normally, the program generates the interrupt start address and branches to that address.

(5) Check whether an interrupt is the watchdog interrupt by reading G0ICR. Check WDIF of G0ICR with the bit test instruction (BTST). Execute the interrupt service routine if WDIF is 1.

> The CPU does not accept other interrupts because the IMn flags of PSW become the highest level during interrupt service routine.

(6) Clear WDIF of G0ICR.

(7) Return to the original program with the RTI instruction after the interrupt service routine ends.

The watchdog timer and the oscillation stabilization wait counter are same. The watchdog timer counts the oscillation stabilization wait when the CPU returns from the STOP mode. Therefore, the WDIF flag of G0ICR is cleared to 0 when the CPU moves to the STOP mode. In addition, the WDIF flag is again cleared to 0 when the CPU moves to the normal mode. Refer to "2-6 Standby Function" in MN102H Series LSI User's Manual.

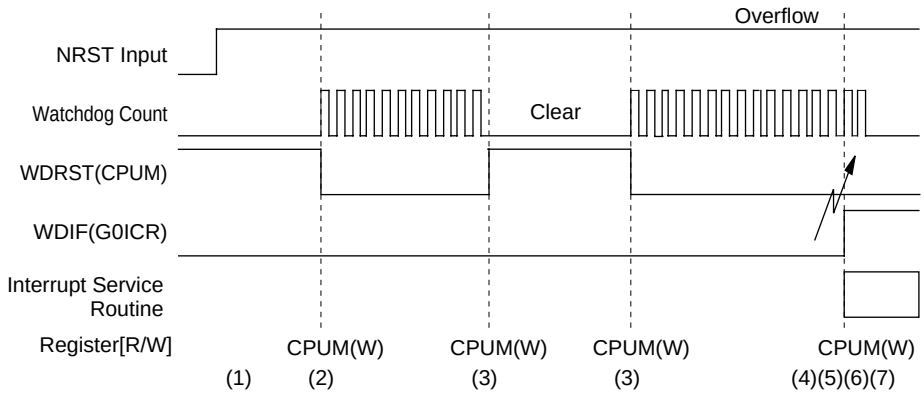


Figure 4-1-14 Watchdog Timer Interrupt Setup Timing

> A watchdog timer does not occur when the CPU releases bus. A watchdog timer occurs when the CPU is in the handshake access wait status.

> An interrupt occurs by ending the bus cycle forcibly when the CPU accepts a watchdog timer interrupt in the handshake access wait status. Therefore, the bus cycle operation cannot be guaranteed when an interrupt occurs during wait status.

■ Watchdog Timer Count Cycle Setup

The WDMD[1:0] flags of the CPUM register set the watchdog timer count cycles.

Setting WDMD[1:0] flags allows the following:

To shorten the interval of the oscillation stabilization wait from the STOP mode.

To shorten the interval of the watchdog interrupt.

WDMD1	WDMD0	Watchdog Interrupt Interval	Oscillation stabilization wait interval
0	0	2^{16} Cycles	$2^{17} \times$ (1/BOSC)
0	1	Reserved	Reserved
1	0	2^{12} Cycles	$2^{13} \times$ (1/BOSC)
1	1	2^{14} Cycles	$2^{15} \times$ (1/BOSC)

4-1-9 Extended Watchdog Timer Setup Example

The MN102H74G/74F/74D/F74G contains the extended watchdog timer which generates a longer watchdog interrupt than a normal watchdog interrupt. The extended watchdog timer also reset the CPU instead of generating an interrupt. In this example, the extended watchdog timer resets the CPU judging an error if the watchdog timer or the extended watchdog timer are not cleared for 5.46 s with a 12-MHz external oscillator.

The CPU runs the same operation at reset when NRST pin is low level. Select the error detect time to 2^{17} BOSC cycles by the CPUM register and 2^{10} cycles by the WDREG register. Because BOSC cycle is approximately 41.6 ns with a 12-MHz external oscillator, the error detect time becomes 41.6 ns $\times$ 2^{10} $\times$ 2^{17} = 5.46 s. During 5.46 s of the error detect time, the watchdog timer and the extended watchdog timer need to be cleared.

■ Extended Watchdog Timer Setup

(1) Set the WDP[2:0] flags of the WDREG register to the error detect time. In this example, set 2^{10}. In addition, set the WDRST flag to 1 because the CPU is reset as soon as a watchdog interrupt occurs.

WDREG: x'00FC5A'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD CLR	-	-	-	-	WD P2	WD P1	WD P0	-	-	-	-	-	-	-	WD RST
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1

The following is WDP[2:0] setting.

000: 1
001: 2^2
010: 2^4
011: 2^6
100: 2^8
101: 2^{10}
110: 2^{12}

(2) Start the watchdog timer and the extended watchdog timer by clearing the WDRST flag of the CPUM register. Set the WDMD[1:0] flags of the CPUM register to the error detect time. In this example, set 2^{17}.

CPUM: x'00FC00'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD RST	WD MD1	WD MD0	-	-	-	-	-	-	-	-	OSC ID	STOP	HALT	OSC1	OSC0
0	0	0	-	-	-	-	-	-	-	-	0	0	0	0	0

If WDMD1 and WDMD0 are '00', a watchdog timer occurs when the watchdog timer counts 2^{17} BOSC cycles (5.461 ms with 12-MHz external oscillator). The following is WDMD[1:0] setting.

00: 2^{17}
01: Reserved
10: 2^{13}
11: 2^{15}

After 5.46 s, a watchdog timer interrupt occurs and the CPU resets.

■ Extended Watchdog Timer Clear

(3) Clear the watchdog timer and the extended watchdog timer by setting the WDCLR flag of the WDREG register and the WDRST flag of the CPUM register to 1 and immediately clearing them to 0. The watchdog timer is cleared while the WDRST flag of the CPUM register is 1. The extended watchdog timer is cleared while the WDCLR flag of the WDREG register is 1. Normally, the watchdog timer and the extended watchdog timer are cleared before a watchdog interrupt occurs.

A watchdog timer does not occur when the CPU releases bus. A watchdog timer occurs when the CPU is in the handshake access wait status.

An interrupt occurs by ending the bus cycle forcibly when the CPU accepts a watchdog timer interrupt in the handshake access wait status. Therefore, the bus cycle operation cannot be guaranteed when an interrupt occurs during wait status.

4-1-10 Interrupt Return

Allocate the instructions such as JMP, JSR and RTS at the end of the interrupt service routine, and branch to the interrupt postprocess. The CPU returns to the main program by interrupt postprocess and hardware process.

(1) Return the register values pushed at the interrupt preprocess during interrupt post process, and then move to hardware process with the RTI instruction.

The following is the example of the interrupt postprocess program. Programming differs depending on user systems. There is no restriction for addresses allocated on the interrupt postprocess program.

```
mov     (10, a3), a0
movx    (2, a3), d1
mov     (a3), d0
mov     d0, mdr
add     14,a3
rti
```

(2) Return the program return address and processor status word from the stack area during hardware process. Hardware executes this process automatically.

Restore the contents of the processor status word (PSW) pushed to the stack (SP).
Restore the contents of the program counter (PC) pushed to the stack (SP+2).
Update the SP data (SP+6 to SP).

> **!** Return IE and IMn to the state before an interrupt is accepted by returning the contents of PSW.

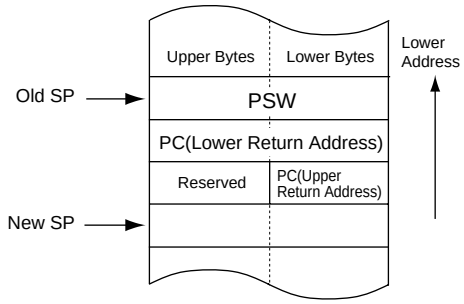


Figure 4-1-15 Stack Pointer (SP) Operation at Interrupt Return

4-1-11 Multiple Interrupt

The CPU disables the interrupt acceptance automatically after it accepts an interrupt. The CPU, however, enables multiple interrupts by program.

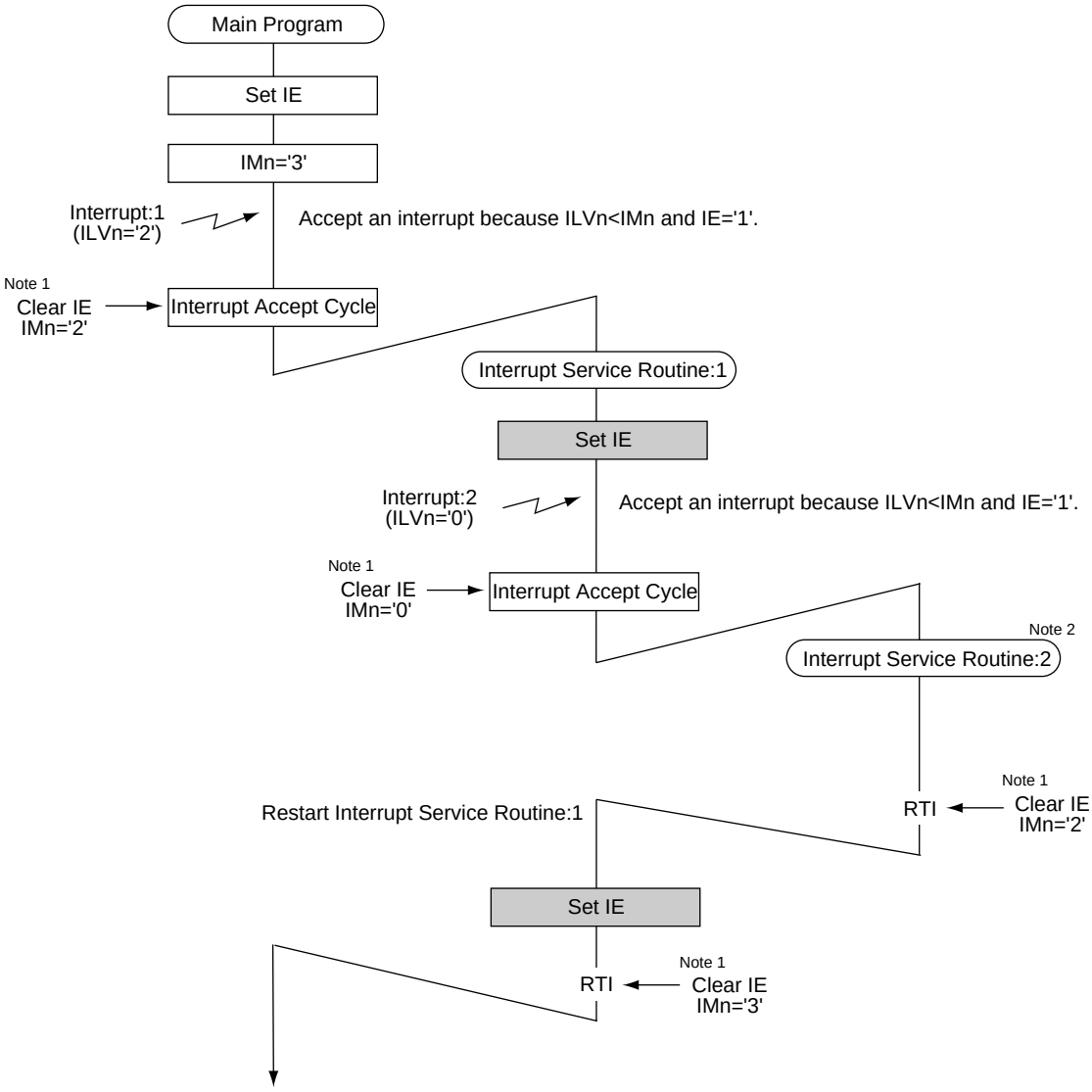
The interrupt enable flag (IE) of PSW is cleared when an interrupt is accepted. This disables all interrupts except a nonmaskable interrupt. The CPU can accept an interrupt with higher priority by writing 1 to the IE flag of PSW during the interrupt service routine.

Enable multiple interrupts with higher level than the mask level (IMn) of PSW.

The stack may overflow although an interrupt with lower priority than the current interrupt is accepted by writing IMn forcibly.

The SP value must be even because the word transfer from odd address is not allowed.

Do not change the maskable interrupt control register (GnICR) while the multiple interrupt is enabled. Clear the IE flag of PSW once when the maskable interrupt control register needs to be changed.



Note 1: The process occurs in hardware.
Note 2: Accept an interrupt with higher priority halting the current interrupt service routine execution because IE is cleared.

Figure 4-1-16 Maskable Interrupt Process Sequence (with Multiple Interrupts)

4-2 Interrupt Control Register

■ Interrupt Accept Group Number Register (IAGR)

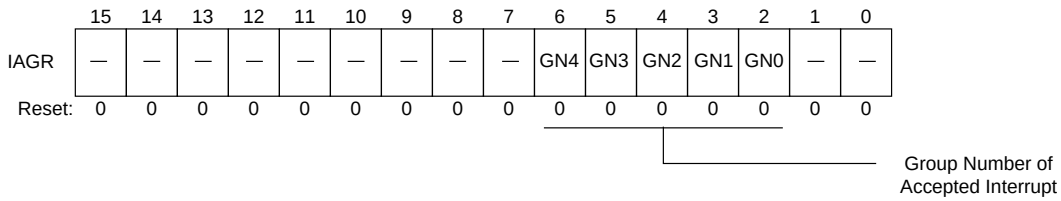
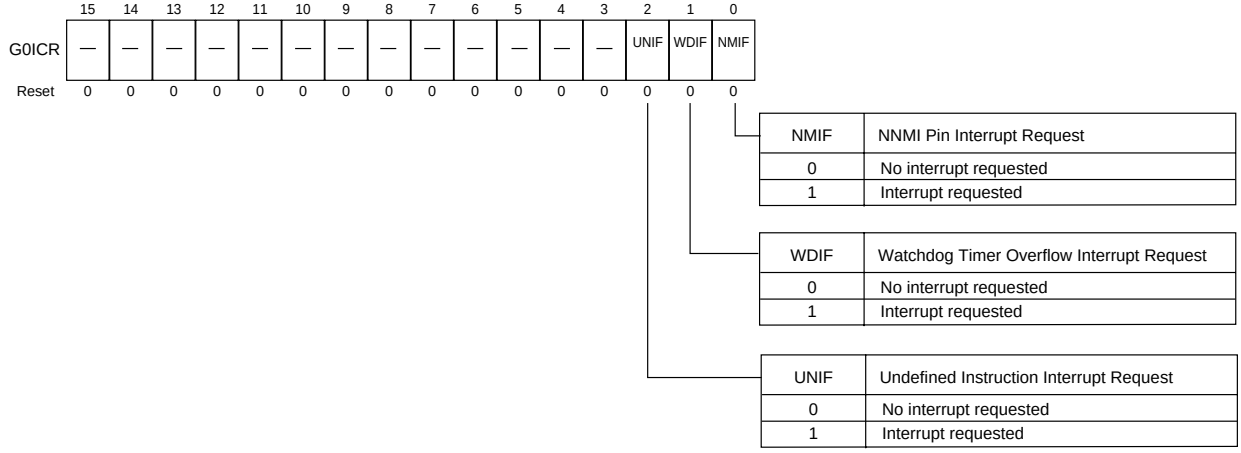


Figure 4-2-1 Interrupt Accept Group Number Register (IAGR: x'00FC0E', R)

This register is read only.

■ Nonmaskable Interrupt Control Register (Group 0) (G0ICR)



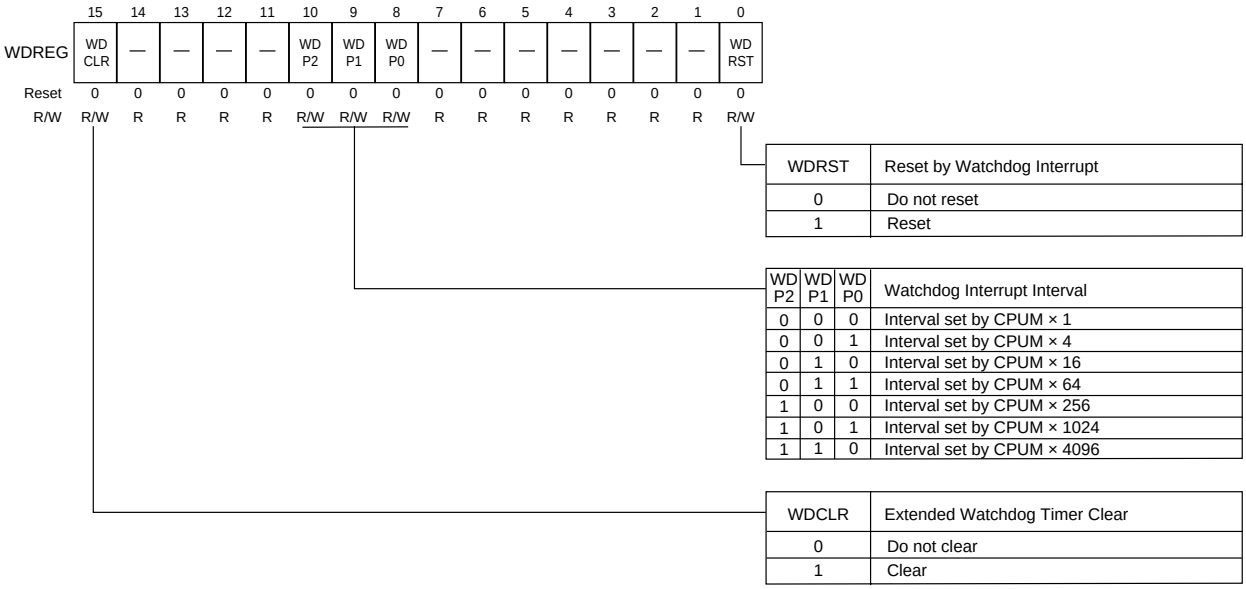
NMIF	NNMI Pin Interrupt Request
0	No interrupt requested
1	Interrupt requested

WDIF	Watchdog Timer Overflow Interrupt Request
0	No interrupt requested
1	Interrupt requested

UNIF	Undefined Instruction Interrupt Request
0	No interrupt requested
1	Interrupt requested

Figure 4-2-2 Nonmaskable Interrupt Control Register (G0ICR: x'00FC40', R/W)

■ Extended Watchdog Interrupt Control Register (WDREG)



WDRST	Reset by Watchdog Interrupt
0	Do not reset
1	Reset

WD P2	WD P1	WD P0	Watchdog Interrupt Interval
0	0	0	Interval set by CPUM × 1
0	0	1	Interval set by CPUM × 4
0	1	0	Interval set by CPUM × 16
0	1	1	Interval set by CPUM × 64
1	0	0	Interval set by CPUM × 256
1	0	1	Interval set by CPUM × 1024
1	1	0	Interval set by CPUM × 4096

WDCLR	Extended Watchdog Timer Clear
0	Do not clear
1	Clear

Figure 4-2-3 Extended Watchdog Interrupt Control Register (WDREG: x'00FC5A', R/W)

■ Maskable Interrupt Control Register (Group 1) (G1ICR)

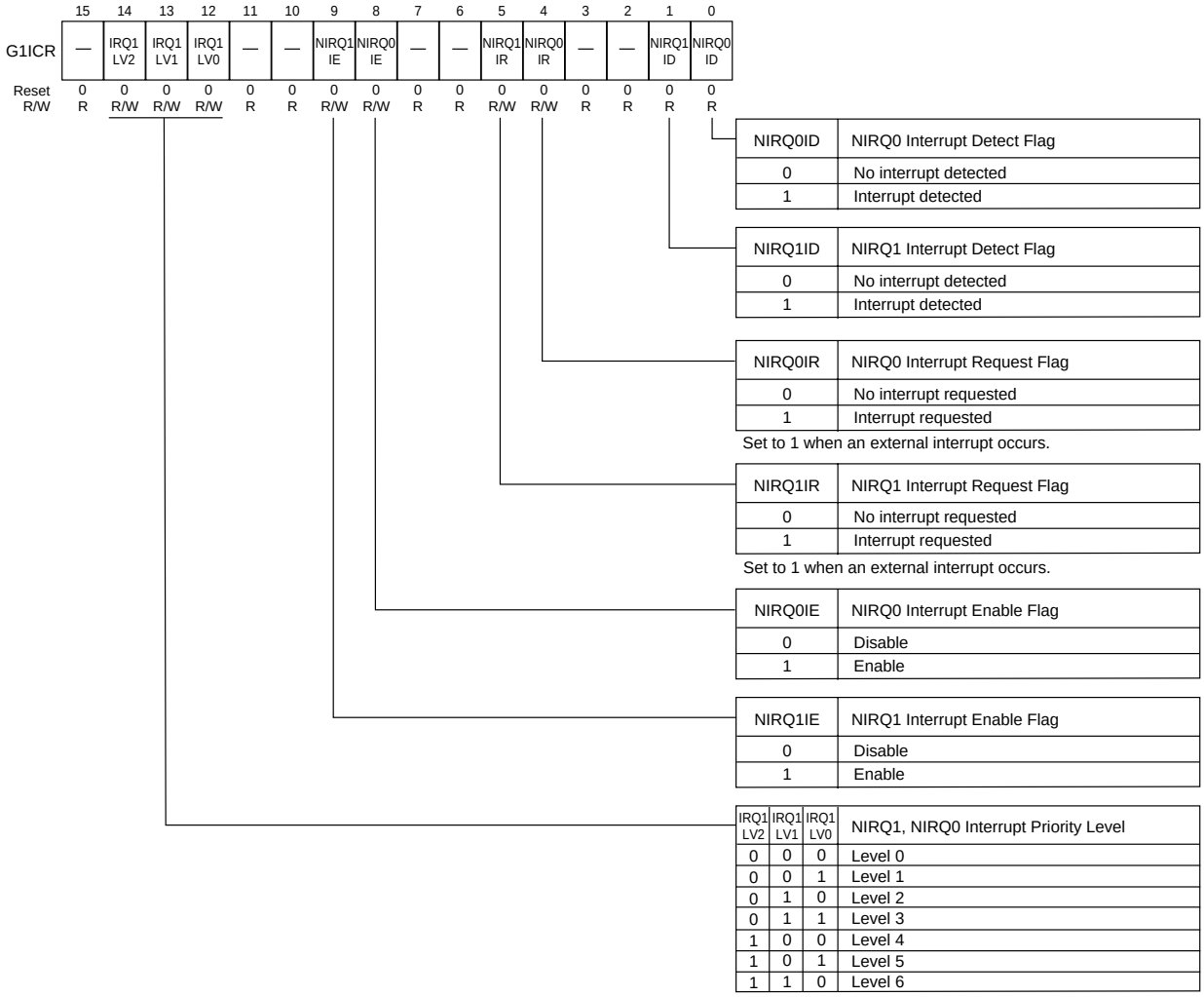

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G1ICR — IRQ1 LV2 IRQ1 LV1 IRQ1 LV0 — — NIRQ1 IE NIRQ0 IE — — NIRQ1 IR NIRQ0 IR — — NIRQ1 ID NIRQ0 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ0ID NIRQ0 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ1ID NIRQ1 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ0IR NIRQ0 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an external interrupt occurs.
NIRQ1IR NIRQ1 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an external interrupt occurs.
NIRQ0IE NIRQ0 Interrupt Enable Flag
0 Disable
1 Enable
NIRQ1IE NIRQ1 Interrupt Enable Flag
0 Disable
1 Enable
IRQ1 LV2 IRQ1 LV1 IRQ1 LV0 NIRQ1, NIRQ0 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-4 Maskable Interrupt Control Register (Group 1) (G1ICR: x'00FC42', R/W)

■ Maskable Interrupt Control Register (Group 2) (G2ICR)

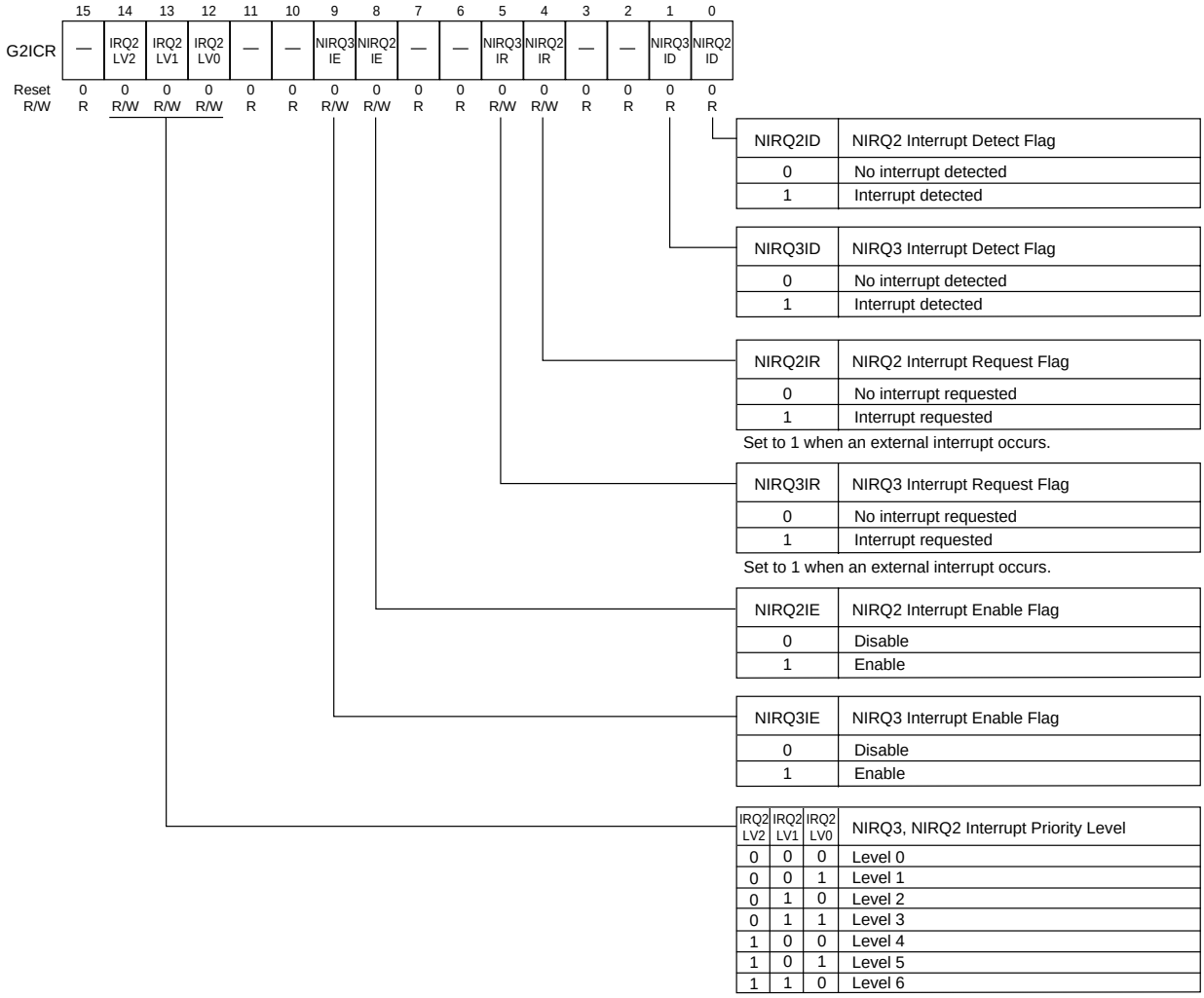

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G2ICR — IRQ2 LV2 IRQ2 LV1 IRQ2 LV0 — — NIRQ3 IE NIRQ2 IE — — NIRQ3 IR NIRQ2 IR — — NIRQ3 ID NIRQ2 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ2ID NIRQ2 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ3ID NIRQ3 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ2IR NIRQ2 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an external interrupt occurs.
NIRQ3IR NIRQ3 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an external interrupt occurs.
NIRQ2IE NIRQ2 Interrupt Enable Flag
0 Disable
1 Enable
NIRQ3IE NIRQ3 Interrupt Enable Flag
0 Disable
1 Enable
IRQ2 LV2 IRQ2 LV1 IRQ2 LV0 NIRQ3, NIRQ2 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-5 Maskable Interrupt Control Register (Group 2) (G2ICR: x'00FC44', R/W)

■ Maskable Interrupt Control Register (Group 3) (G3ICR)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3ICR	—	IRQ3 LV2	IRQ3 LV1	IRQ3 LV0	—	—	NIRQ5 IE	NIRQ4 IE	—	—	NIRQ5 IR	NIRQ4 IR	—	—	NIRQ5 ID	NIRQ4 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R

NIRQ4ID	NIRQ4 Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

NIRQ5ID	NIRQ5 Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

NIRQ4IR	NIRQ4 Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an external interrupt occurs.

NIRQ5IR	NIRQ5 Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an external interrupt occurs.

NIRQ4IE	NIRQ4 Interrupt Enable Flag
0	Disable
1	Enable

NIRQ5IE	NIRQ5 Interrupt Enable Flag
0	Disable
1	Enable

IRQ3 LV2	IRQ3 LV1	IRQ3 LV0	NIRQ5, NIRQ4 Interrupt Priority Level
0	0	0	Level 0
0	0	1	Level 1
0	1	0	Level 2
0	1	1	Level 3
1	0	0	Level 4
1	0	1	Level 5
1	1	0	Level 6

Figure 4-2-6 Maskable Interrupt Control Register (Group 3) (G3ICR: x'00FC46', R/W)

■ Maskable Interrupt Control Register (Group 4) (G4ICR)

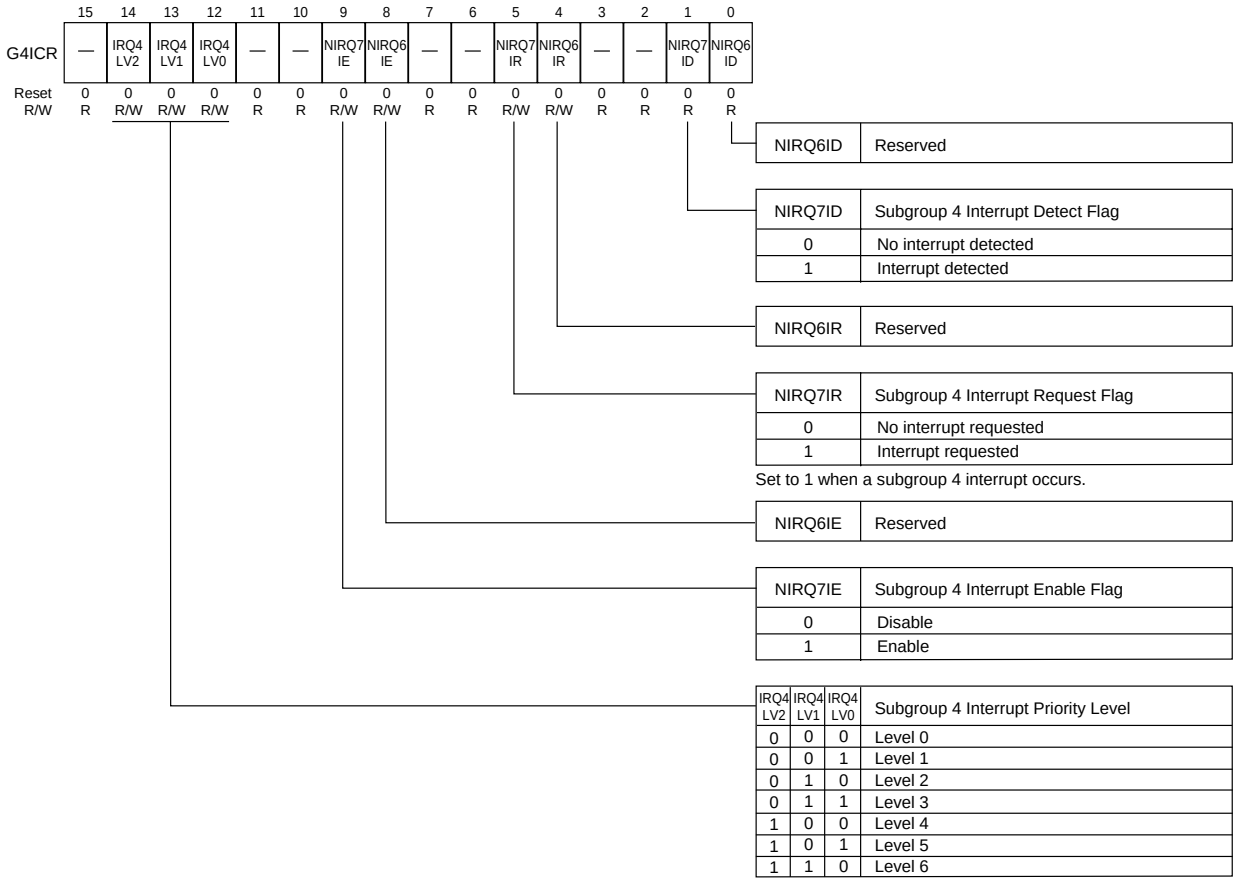

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G4ICR — IRQ4 LV2 IRQ4 LV1 IRQ4 LV0 — — NIRQ7 IE NIRQ6 IE — — NIRQ7 IR NIRQ6 IR — — NIRQ7 ID NIRQ6 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ6ID Reserved
NIRQ7ID Subgroup 4 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ6IR Reserved
NIRQ7IR Subgroup 4 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a subgroup 4 interrupt occurs.
NIRQ6IE Reserved
NIRQ7IE Subgroup 4 Interrupt Enable Flag
0 Disable
1 Enable
IRQ4 LV2 IRQ4 LV1 IRQ4 LV0 Subgroup 4 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-7 Maskable Interrupt Control Register (Group 4) (G4ICR: x'00FC48', R/W)

■ Maskable Interrupt Control Register (Group 5) (G5ICR)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G5ICR	—	IRQ5 LV2	IRQ5 LV1	IRQ5 LV0	—	—	NIRQ9 IE	NIRQ8 IE	—	—	NIRQ9 IR	NIRQ8 IR	—	—	NIRQ9 ID	NIRQ8 ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R

NIRQ8ID	Reserved

NIRQ9ID	Subgroup 5 Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

NIRQ8IR	Reserved

NIRQ9IR	Subgroup 5 Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when a subgroup 5 interrupt occurs.

NIRQ8IE	Reserved

NIRQ9IE	Subgroup 5 Interrupt Enable Flag
0	Disable
1	Enable

IRQ5 LV2	IRQ5 LV1	IRQ5 LV0	Subgroup 5 Interrupt Priority Level
0	0	0	Level 0
0	0	1	Level 1
0	1	0	Level 2
0	1	1	Level 3
1	0	0	Level 4
1	0	1	Level 5
1	1	0	Level 6

Figure 4-2-8 Maskable Interrupt Control Register (Group 5) (G5ICR: x'00FC4A', R/W)

■ Maskable Interrupt Control Register (Group 6) (G6ICR)

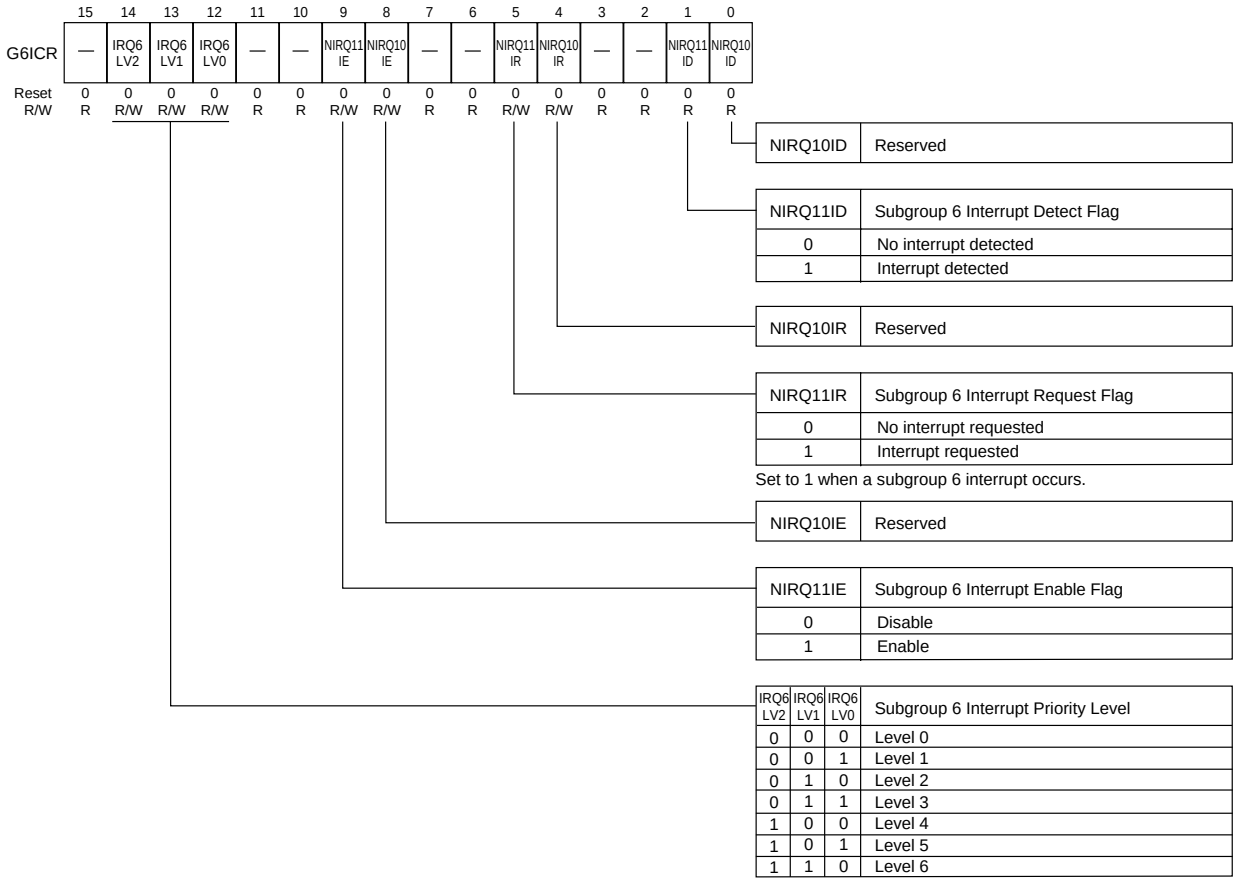

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G6ICR — IRQ6 LV2 IRQ6 LV1 IRQ6 LV0 — — NIRQ11 IE NIRQ10 IE — — NIRQ11 IR NIRQ10 IR — — NIRQ11 ID NIRQ10 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ10ID Reserved
NIRQ11ID Subgroup 6 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ10IR Reserved
NIRQ11IR Subgroup 6 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a subgroup 6 interrupt occurs.
NIRQ10IE Reserved
NIRQ11IE Subgroup 6 Interrupt Enable Flag
0 Disable
1 Enable
IRQ6 LV2 IRQ6 LV1 IRQ6 LV0 Subgroup 6 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-9 Maskable Interrupt Control Register (Group 6) (G6ICR: x'00FC4C', R/W)

■ Maskable Interrupt Control Register (Group 7) (G7ICR)

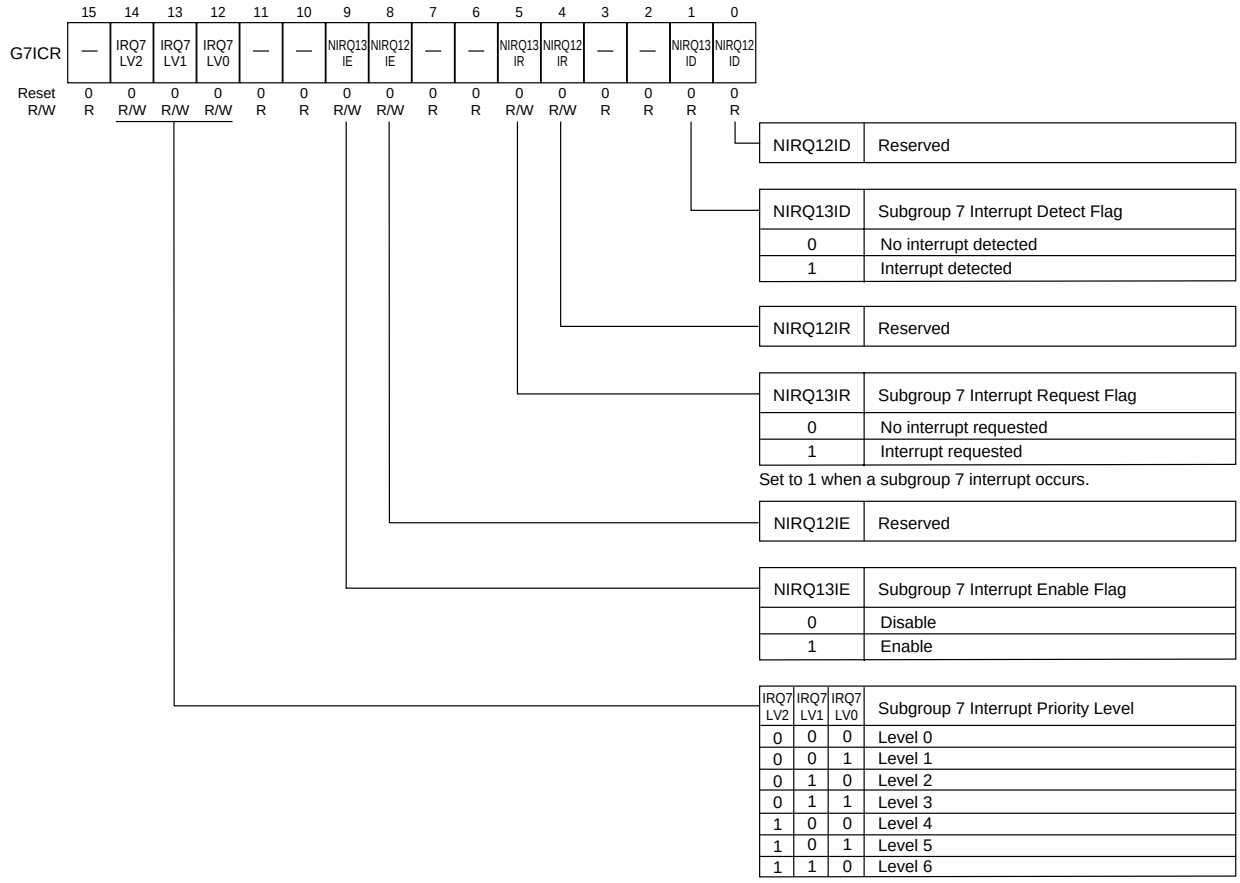

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G7ICR — IRQ7 LV2 IRQ7 LV1 IRQ7 LV0 — — NIRQ13 IE NIRQ12 IE — — NIRQ13 IR NIRQ12 IR — — NIRQ13 ID NIRQ12 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ12ID Reserved
NIRQ13ID Subgroup 7 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ12IR Reserved
NIRQ13IR Subgroup 7 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a subgroup 7 interrupt occurs.
NIRQ12IE Reserved
NIRQ13IE Subgroup 7 Interrupt Enable Flag
0 Disable
1 Enable
IRQ7 LV2 IRQ7 LV1 IRQ7 LV0 Subgroup 7 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-10 Maskable Interrupt Control Register (Group 7) (G7ICR: x'00FC4E', R/W)

■ Maskable Interrupt Control Register (Group 8) (G8ICR)

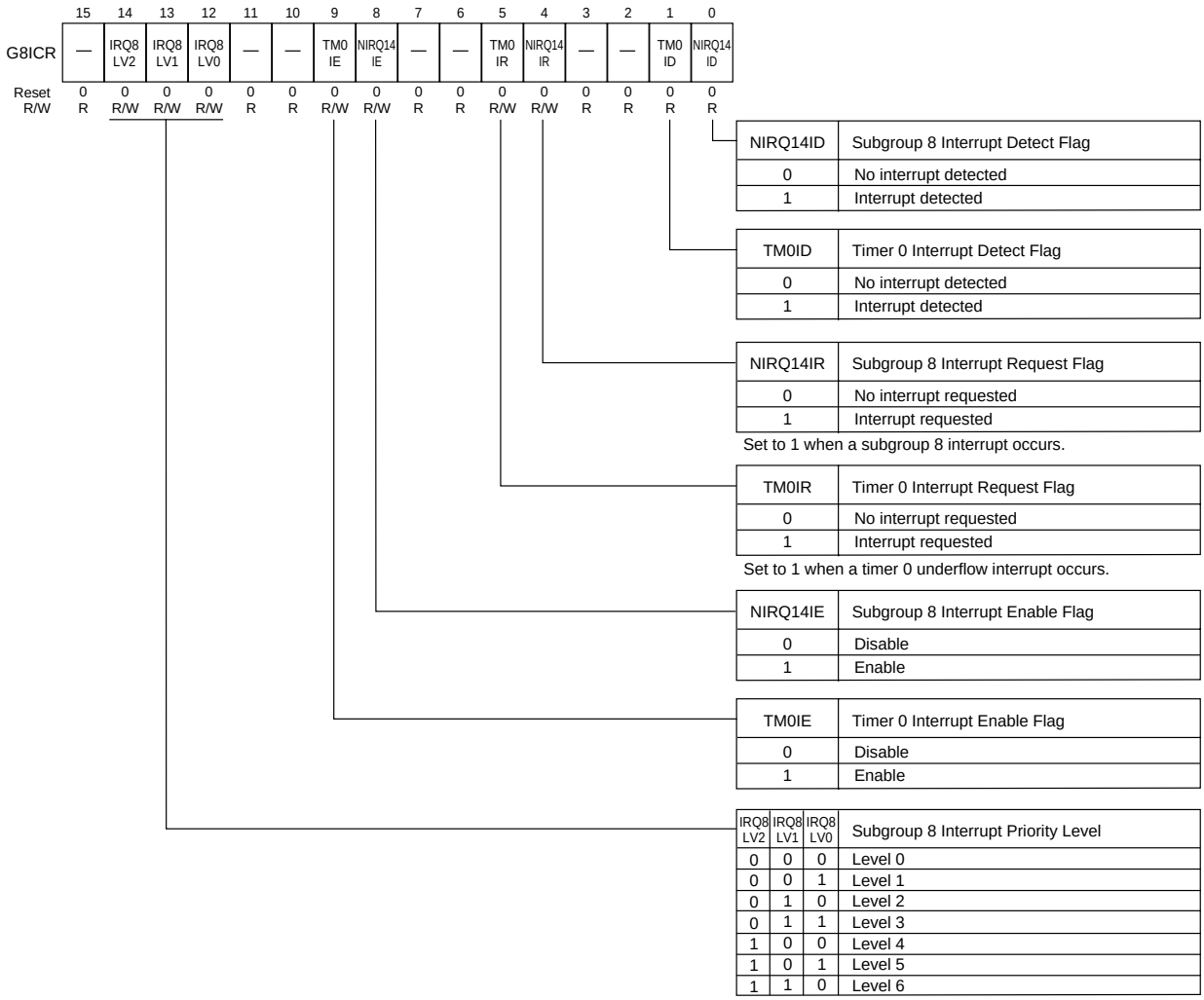

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G8ICR — IRQ8 LV2 IRQ8 LV1 IRQ8 LV0 — — TM0 IE NIRQ14 IE — — TM0 IR NIRQ14 IR — — TM0 ID NIRQ14 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ14ID Subgroup 8 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM0ID Timer 0 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ14IR Subgroup 8 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a subgroup 8 interrupt occurs.
TM0IR Timer 0 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a timer 0 underflow interrupt occurs.
NIRQ14IE Subgroup 8 Interrupt Enable Flag
0 Disable
1 Enable
TM0IE Timer 0 Interrupt Enable Flag
0 Disable
1 Enable
IRQ8 LV2 IRQ8 LV1 IRQ8 LV0 Subgroup 8 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-11 Maskable Interrupt Control Register (Group 8) (G8ICR: x'00FC50', R/W)

■ Maskable Interrupt Control Register (Group 9) (G9ICR)

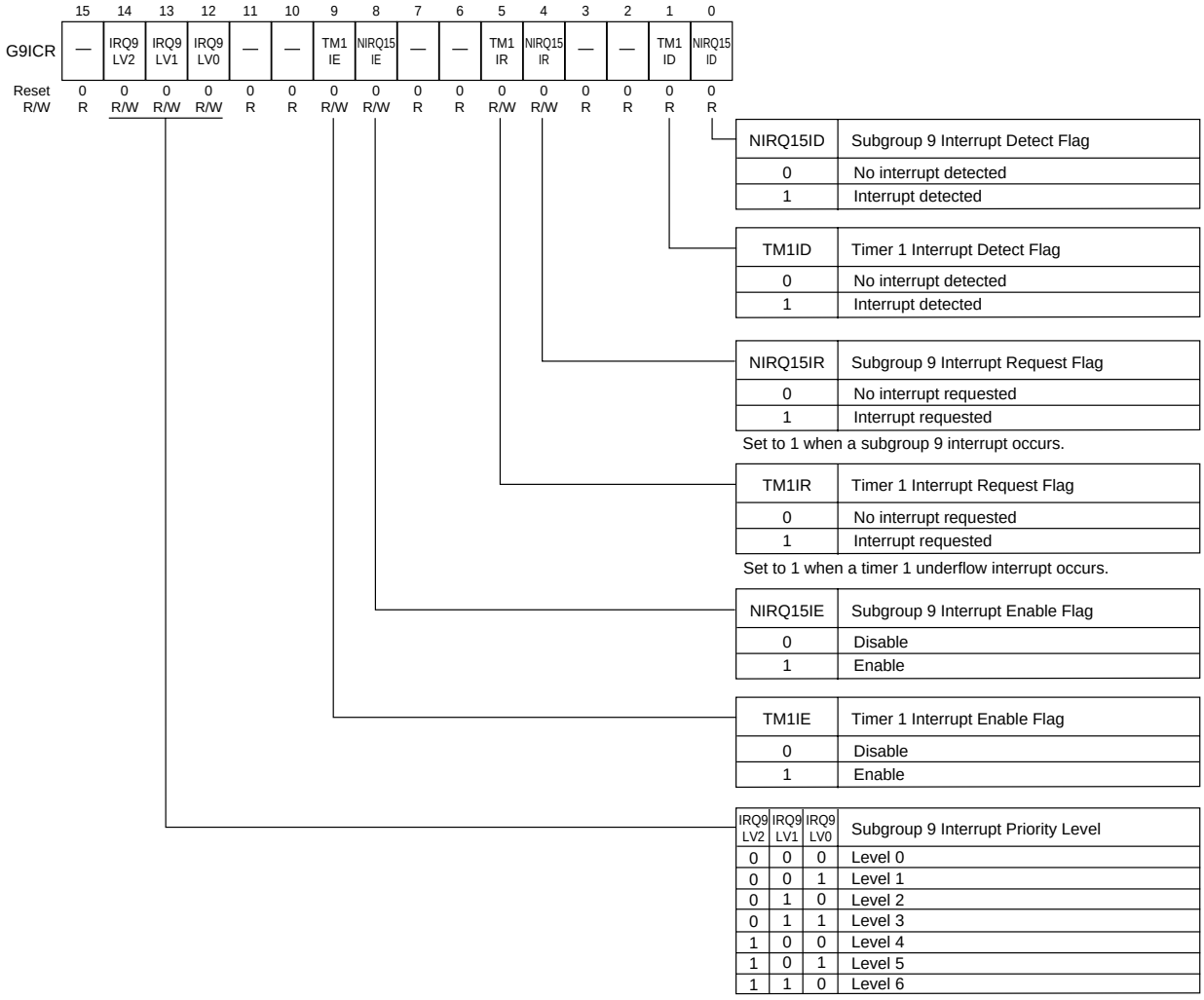

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
G9ICR — IRQ9 LV2 IRQ9 LV1 IRQ9 LV0 — — TM1 IE NIRQ15 IE — — TM1 IR NIRQ15 IR — — TM1 ID NIRQ15 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R/W R/W R/W R R R/W R/W R R R/W R/W R R R R
NIRQ15ID Subgroup 9 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM1ID Timer 1 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
NIRQ15IR Subgroup 9 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a subgroup 9 interrupt occurs.
TM1IR Timer 1 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when a timer 1 underflow interrupt occurs.
NIRQ15IE Subgroup 9 Interrupt Enable Flag
0 Disable
1 Enable
TM1IE Timer 1 Interrupt Enable Flag
0 Disable
1 Enable
IRQ9 LV2 IRQ9 LV1 IRQ9 LV0 Subgroup 9 Interrupt Priority Level
0 0 0 Level 0
0 0 1 Level 1
0 1 0 Level 2
0 1 1 Level 3
1 0 0 Level 4
1 0 1 Level 5
1 1 0 Level 6

Figure 4-2-12 Maskable Interrupt Control Register (Group 9) (G9ICR: x'00FC52', R/W)

■ Subgroup Maskable Interrupt Control Register 40 (SG40ICR)

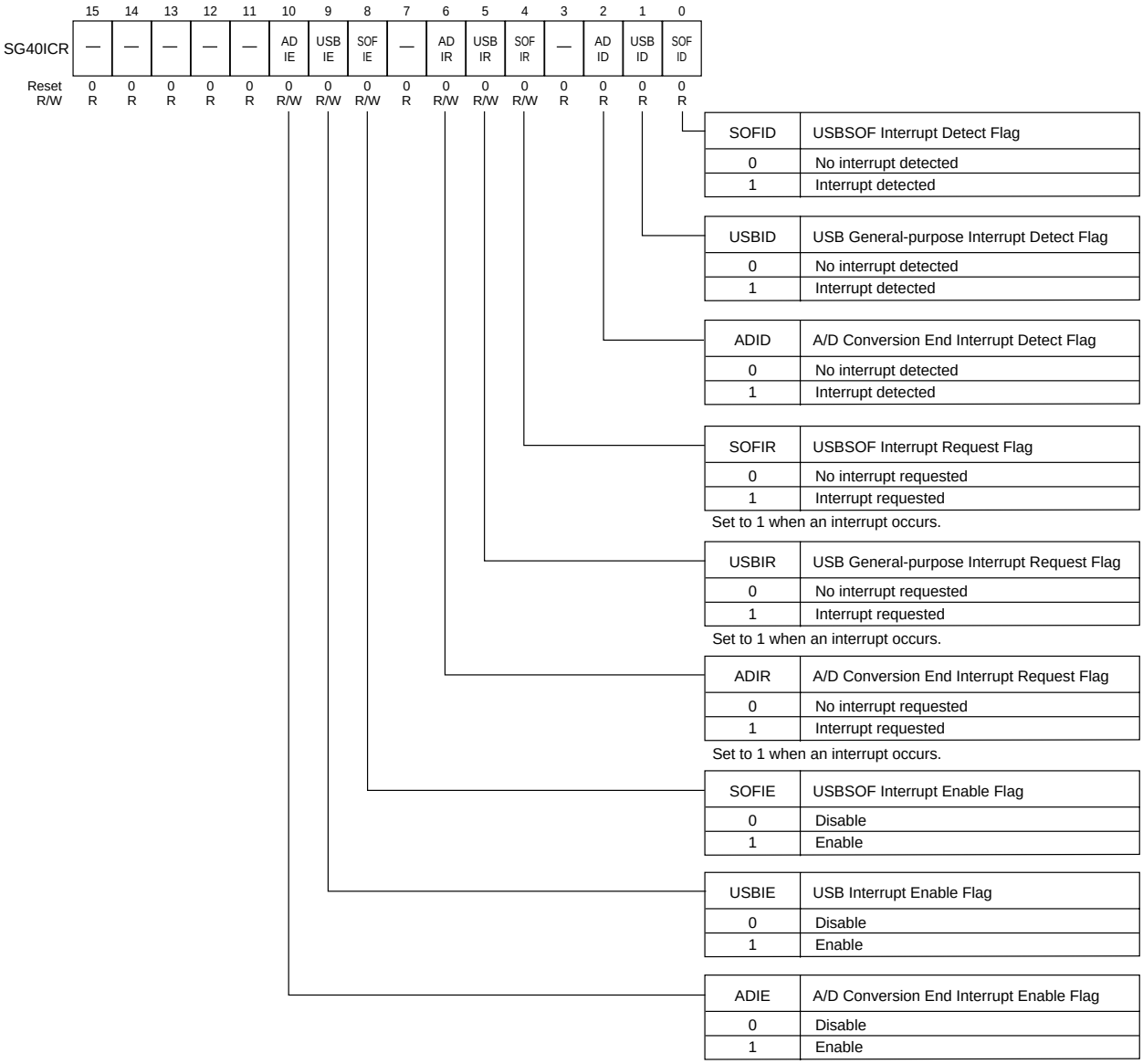
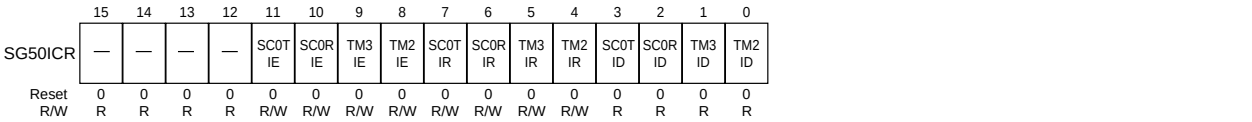

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG40ICR — — — — — AD IE USB IE SOF IE — AD IR USB IR SOF IR — AD ID USB ID SOF ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R R/W R/W R/W R R/W R/W R/W R R R R
SOFID USBSOF Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
USBID USB General-purpose Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
ADID A/D Conversion End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SOFIR USBSOF Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
USBIR USB General-purpose Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
ADIR A/D Conversion End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SOFIE USBSOF Interrupt Enable Flag
0 Disable
1 Enable
USBIE USB Interrupt Enable Flag
0 Disable
1 Enable
ADIE A/D Conversion End Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-13 Subgroup Maskable Interrupt Control Register 40 (SG40ICR: x'E00600', R/W)

■ Subgroup Maskable Interrupt Control Register 50 (SG50ICR)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG50ICR	—	—	—	—	SC0TIE	SC0RIE	TM3IE	TM2IE	SC0TIR	SC0RIR	TM3IR	TM2IR	SC0TID	SC0RID	TM3ID	TM2ID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

TM2ID	Timer 2 Underflow Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

TM3ID	Timer 3 Underflow Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

SC0RID	Serial 0 Reception End Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

SC0TID	Serial 0 Transmission End Interrupt Detect Flag
0	No interrupt detected
1	Interrupt detected

TM2IR	Timer 2 Underflow Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an interrupt occurs.

TM3IR	Timer 3 Underflow Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an interrupt occurs.

SC0RIR	Serial 0 Reception End Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an interrupt occurs.

SC0TIR	Serial 0 Transmission End Interrupt Request Flag
0	No interrupt requested
1	Interrupt requested

Set to 1 when an interrupt occurs.

TM2IE	Timer 2 Underflow Interrupt Enable Flag
0	Disable
1	Enable

TM3IE	Timer 3 Underflow Interrupt Enable Flag
0	Disable
1	Enable

SC0RIE	Serial 0 Reception End Interrupt Enable Flag
0	Disable
1	Enable

SC0TIE	Serial 0 Transmission End Interrupt Enable Flag
0	Disable
1	Enable

Figure 4-2-14 Subgroup Maskable Interrupt Control Register 50 (SG50ICR: x'E00602', R/W)

■ Subgroup Maskable Interrupt Control Register 60 (SG60ICR)

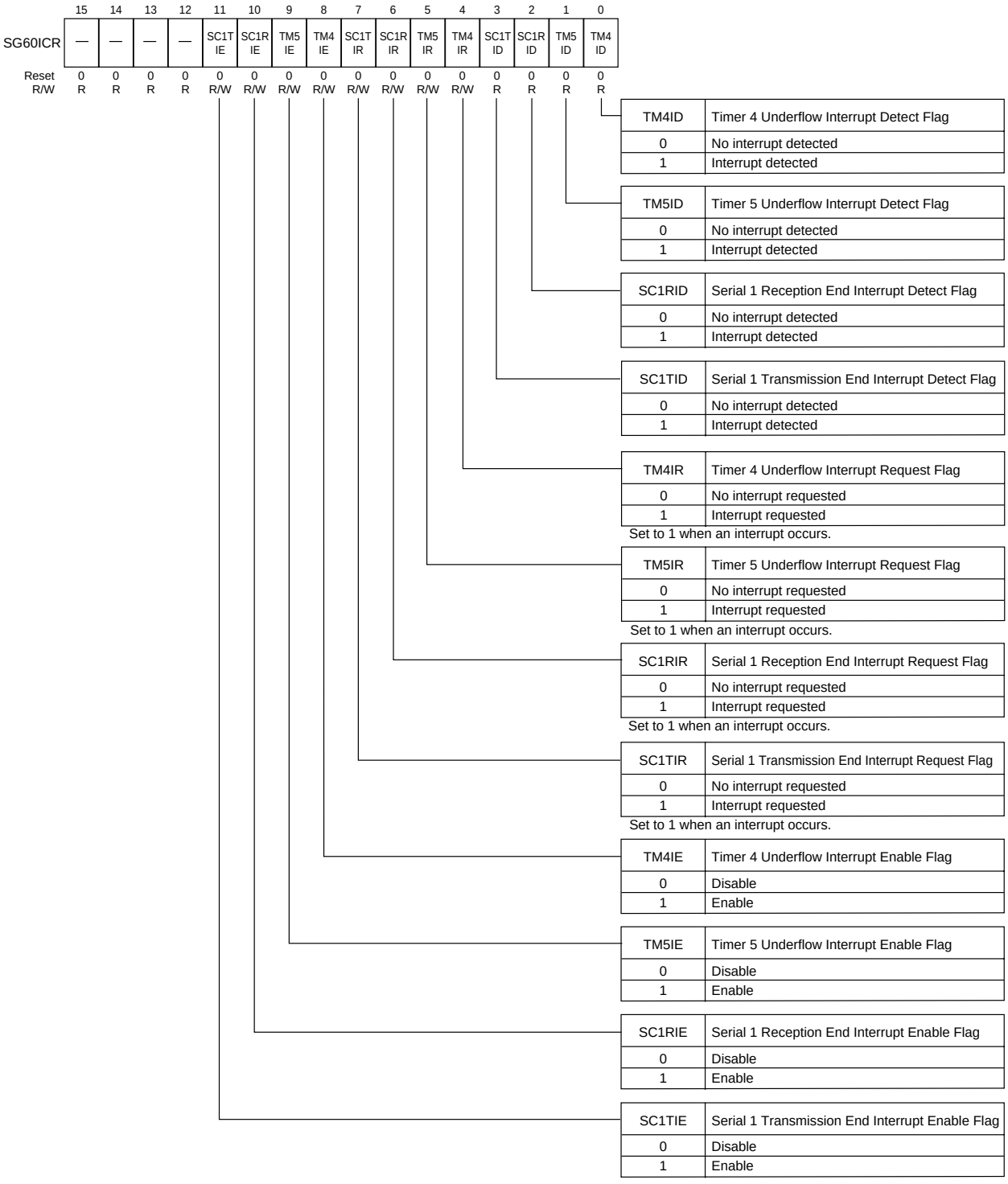

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG60ICR — — — — SC1TIE SC1RIE TM5IE TM4IE SC1TIR SC1RIR TM5IR TM4IR SC1TID SC1RID TM5ID TM4ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
TM4ID Timer 4 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM5ID Timer 5 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC1RID Serial 1 Reception End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC1TID Serial 1 Transmission End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM4IR Timer 4 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM5IR Timer 5 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC1RIR Serial 1 Reception End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC1TIR Serial 1 Transmission End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM4IE Timer 4 Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM5IE Timer 5 Underflow Interrupt Enable Flag
0 Disable
1 Enable
SC1RIE Serial 1 Reception End Interrupt Enable Flag
0 Disable
1 Enable
SC1TIE Serial 1 Transmission End Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-15 Subgroup Maskable Interrupt Control Register 60 (SG60ICR: x'E00604', R/W)

■ Subgroup Maskable Interrupt Control Register 61 (SG61ICR)

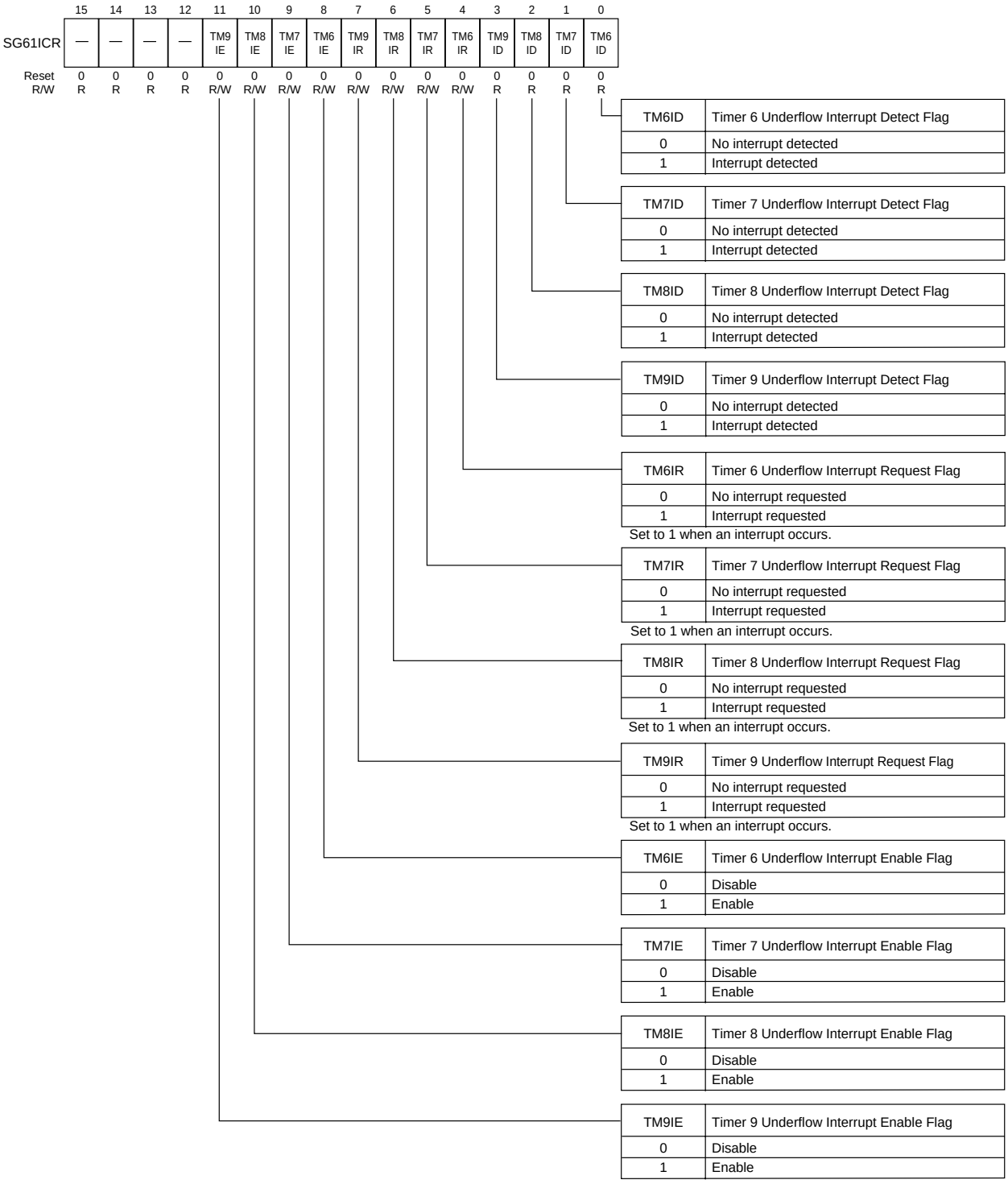

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG61ICR — — — — TM9 IE TM8 IE TM7 IE TM6 IE TM9 IR TM8 IR TM7 IR TM6 IR TM9 ID TM8 ID TM7 ID TM6 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
TM6ID Timer 6 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM7ID Timer 7 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM8ID Timer 8 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM9ID Timer 9 Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM6IR Timer 6 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM7IR Timer 7 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM8IR Timer 8 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM9IR Timer 9 Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM6IE Timer 6 Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM7IE Timer 7 Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM8IE Timer 8 Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM9IE Timer 9 Underflow Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-16 Subgroup Maskable Interrupt Control Register 61 (SG61ICR: x'E00606', R/W)

■ Subgroup Maskable Interrupt Control Register 70 (SG70ICR)

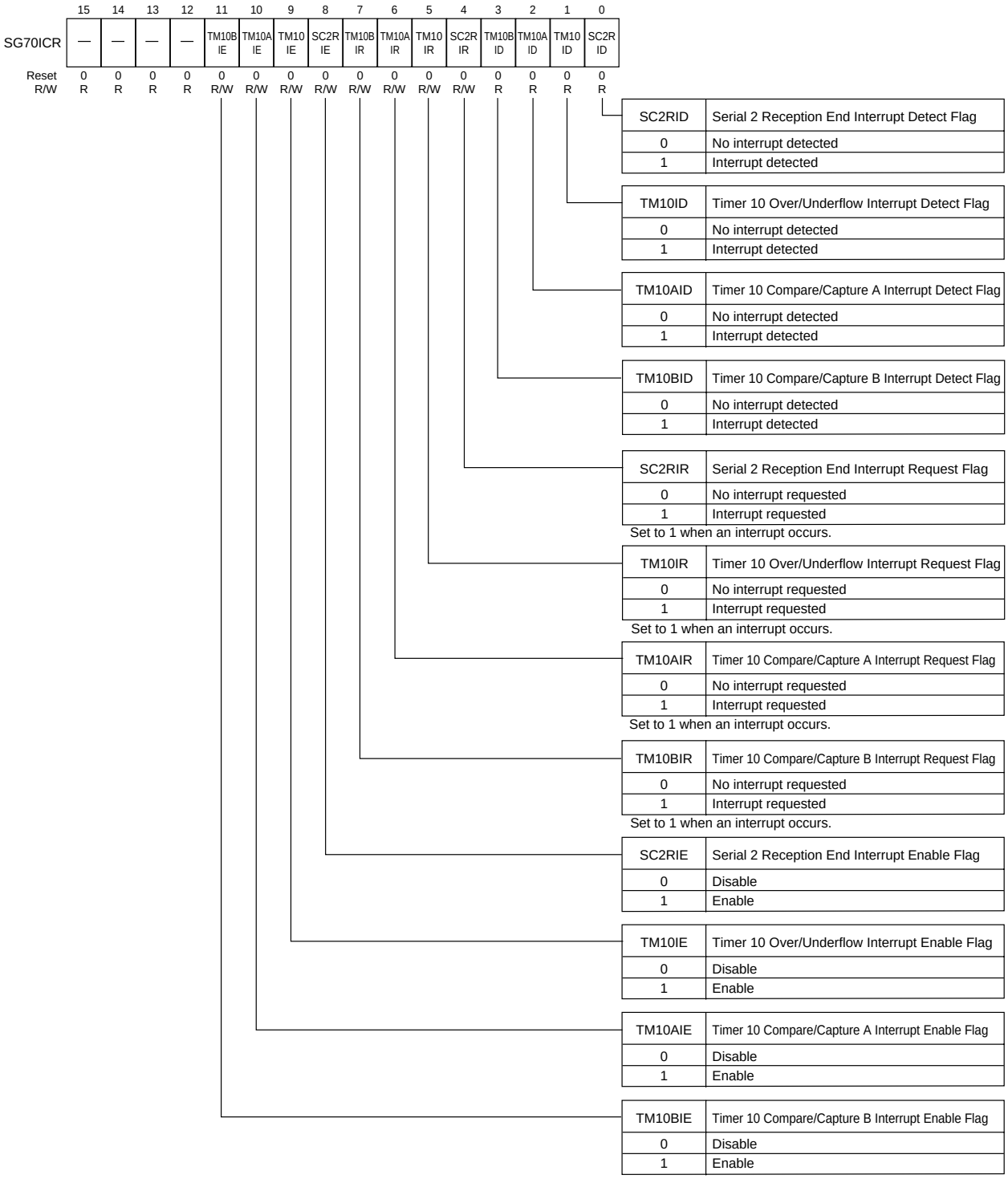

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG70ICR — — — — TM10BIE TM10AIE TM10IE SC2RIE TM10BIR TM10AIR TM10IR SC2RIR TM10BID TM10AID TM10ID SC2RID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
SC2RID Serial 2 Reception End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM10ID Timer 10 Over/Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM10AID Timer 10 Compare/Capture A Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM10BID Timer 10 Compare/Capture B Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC2RIR Serial 2 Reception End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM10IR Timer 10 Over/Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM10AIR Timer 10 Compare/Capture A Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM10BIR Timer 10 Compare/Capture B Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC2RIE Serial 2 Reception End Interrupt Enable Flag
0 Disable
1 Enable
TM10IE Timer 10 Over/Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM10AIE Timer 10 Compare/Capture A Interrupt Enable Flag
0 Disable
1 Enable
TM10BIE Timer 10 Compare/Capture B Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-17 Subgroup Maskable Interrupt Control Register 70 (SG70ICR: x'E00608', R/W)

■ Subgroup Maskable Interrupt Control Register 71 (SG71ICR)

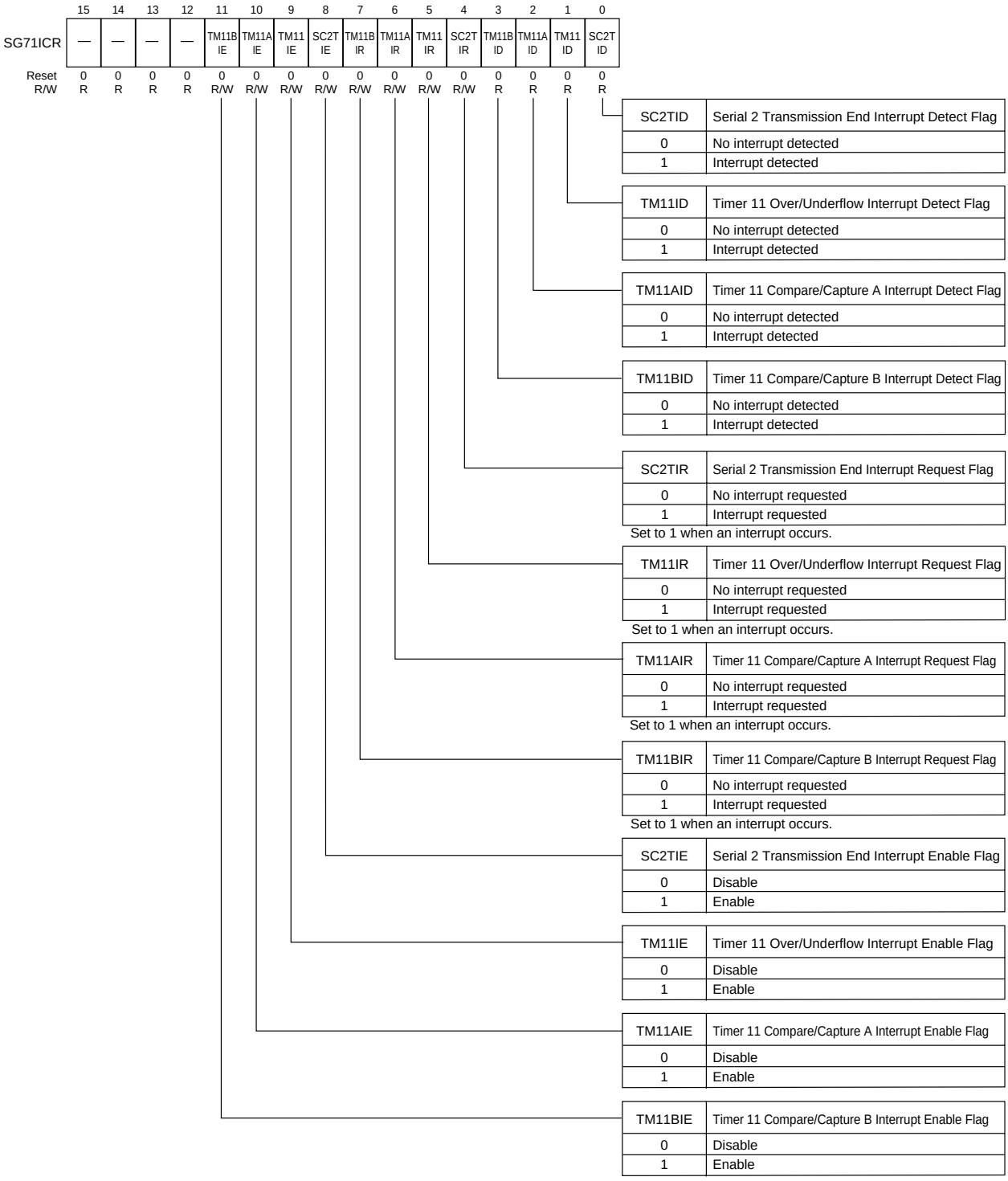

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG71ICR — — — — TM11BIE TM11AIE TM11IE SC2TIE TM11BIR TM11AIR TM11IR SC2TIR TM11BID TM11AID TM11ID SC2TID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
SC2TID Serial 2 Transmission End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM11ID Timer 11 Over/Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM11AID Timer 11 Compare/Capture A Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM11BID Timer 11 Compare/Capture B Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC2TIR Serial 2 Transmission End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM11IR Timer 11 Over/Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM11AIR Timer 11 Compare/Capture A Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM11BIR Timer 11 Compare/Capture B Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC2TIE Serial 2 Transmission End Interrupt Enable Flag
0 Disable
1 Enable
TM11IE Timer 11 Over/Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM11AIE Timer 11 Compare/Capture A Interrupt Enable Flag
0 Disable
1 Enable
TM11BIE Timer 11 Compare/Capture B Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-18 Subgroup Maskable Interrupt Control Register 71 (SG71ICR: x'E0060A', R/W)

■ Subgroup Maskable Interrupt Control Register 80 (SG80ICR)

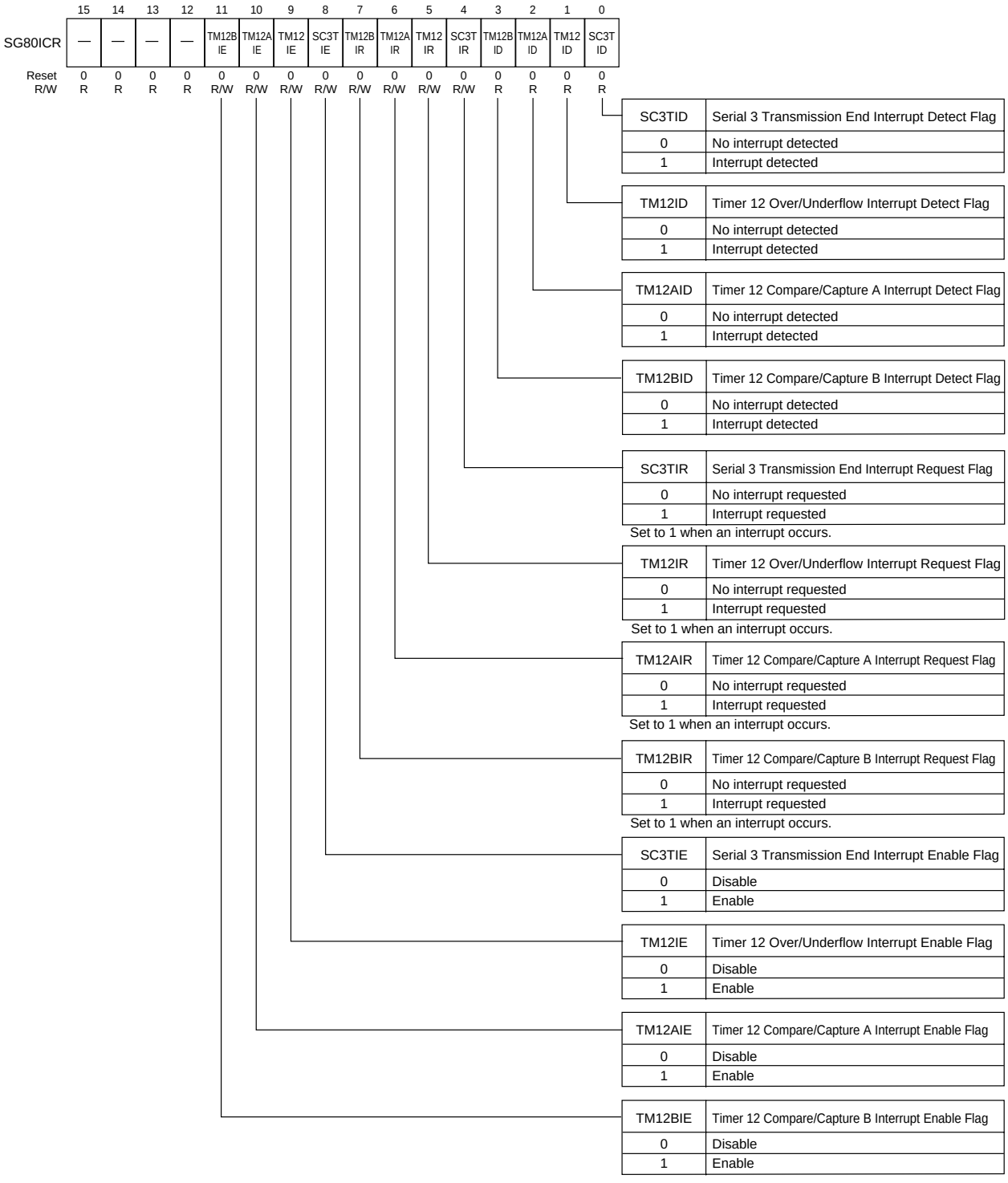

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG80ICR — — — — TM12BIE TM12AIE TM12IE SC3TIE TM12BIR TM12AIR TM12IR SC3TIR TM12BID TM12AID TM12ID SC3TID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
SC3TID Serial 3 Transmission End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM12ID Timer 12 Over/Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM12AID Timer 12 Compare/Capture A Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM12BID Timer 12 Compare/Capture B Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC3TIR Serial 3 Transmission End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM12IR Timer 12 Over/Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM12AIR Timer 12 Compare/Capture A Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM12BIR Timer 12 Compare/Capture B Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC3TIE Serial 3 Transmission End Interrupt Enable Flag
0 Disable
1 Enable
TM12IE Timer 12 Over/Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM12AIE Timer 12 Compare/Capture A Interrupt Enable Flag
0 Disable
1 Enable
TM12BIE Timer 12 Compare/Capture B Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-19 Subgroup Maskable Interrupt Control Register 80 (SG80ICR: x'E0060C', R/W)

■ Subgroup Maskable Interrupt Control Register 81 (SG81ICR)

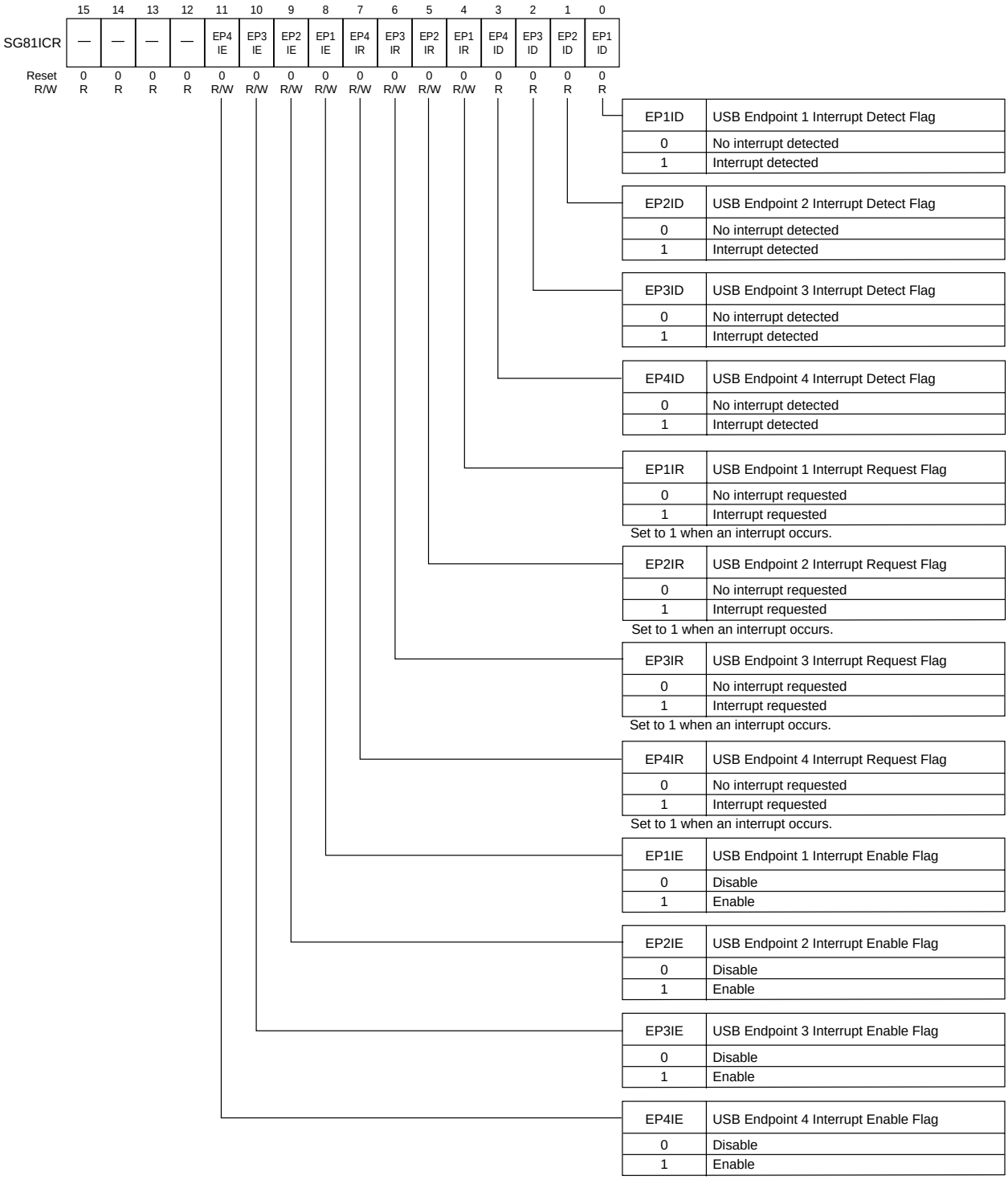

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG81ICR — — — — EP4 IE EP3 IE EP2 IE EP1 IE EP4 IR EP3 IR EP2 IR EP1 IR EP4 ID EP3 ID EP2 ID EP1 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
EP1ID USB Endpoint 1 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP2ID USB Endpoint 2 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP3ID USB Endpoint 3 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP4ID USB Endpoint 4 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP1IR USB Endpoint 1 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP2IR USB Endpoint 2 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP3IR USB Endpoint 3 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP4IR USB Endpoint 4 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP1IE USB Endpoint 1 Interrupt Enable Flag
0 Disable
1 Enable
EP2IE USB Endpoint 2 Interrupt Enable Flag
0 Disable
1 Enable
EP3IE USB Endpoint 3 Interrupt Enable Flag
0 Disable
1 Enable
EP4IE USB Endpoint 4 Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-20 Subgroup Maskable Interrupt Control Register 81 (SG81ICR: x'E0061C', R/W)

■ Subgroup Maskable Interrupt Control Register 90 (SG90ICR)

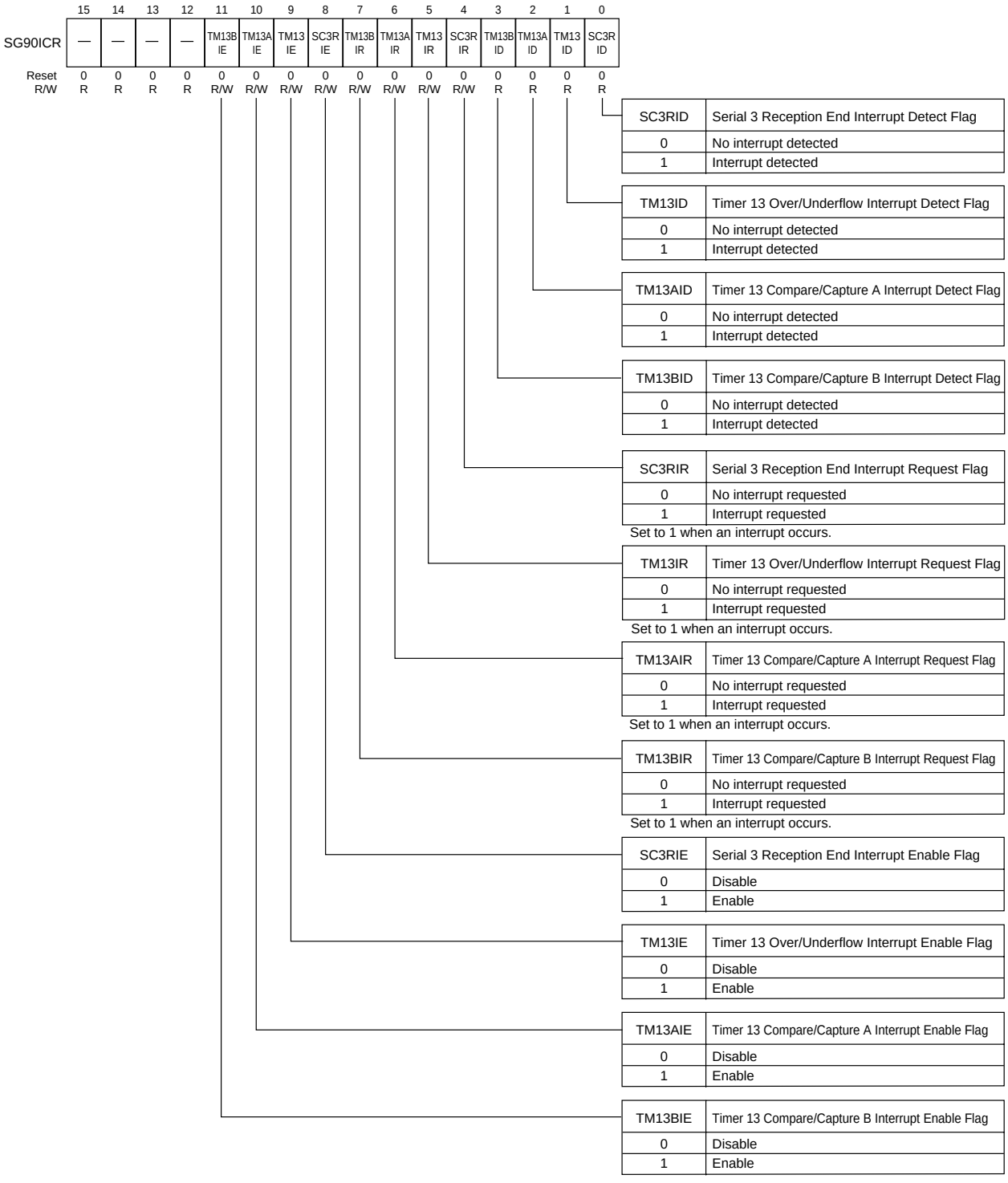

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG90ICR — — — — TM13BIE TM13AIE TM13IE SC3RIE TM13BIR TM13AIR TM13IR SC3RIR TM13BID TM13AID TM13ID SC3RID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
SC3RID Serial 3 Reception End Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM13ID Timer 13 Over/Underflow Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM13AID Timer 13 Compare/Capture A Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
TM13BID Timer 13 Compare/Capture B Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
SC3RIR Serial 3 Reception End Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM13IR Timer 13 Over/Underflow Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM13AIR Timer 13 Compare/Capture A Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
TM13BIR Timer 13 Compare/Capture B Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
SC3RIE Serial 3 Reception End Interrupt Enable Flag
0 Disable
1 Enable
TM13IE Timer 13 Over/Underflow Interrupt Enable Flag
0 Disable
1 Enable
TM13AIE Timer 13 Compare/Capture A Interrupt Enable Flag
0 Disable
1 Enable
TM13BIE Timer 13 Compare/Capture B Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-21 Subgroup Maskable Interrupt Control Register 90 (SG90ICR: x'E0060E', R/W)

■ Subgroup Maskable Interrupt Control Register 91 (SG91ICR)

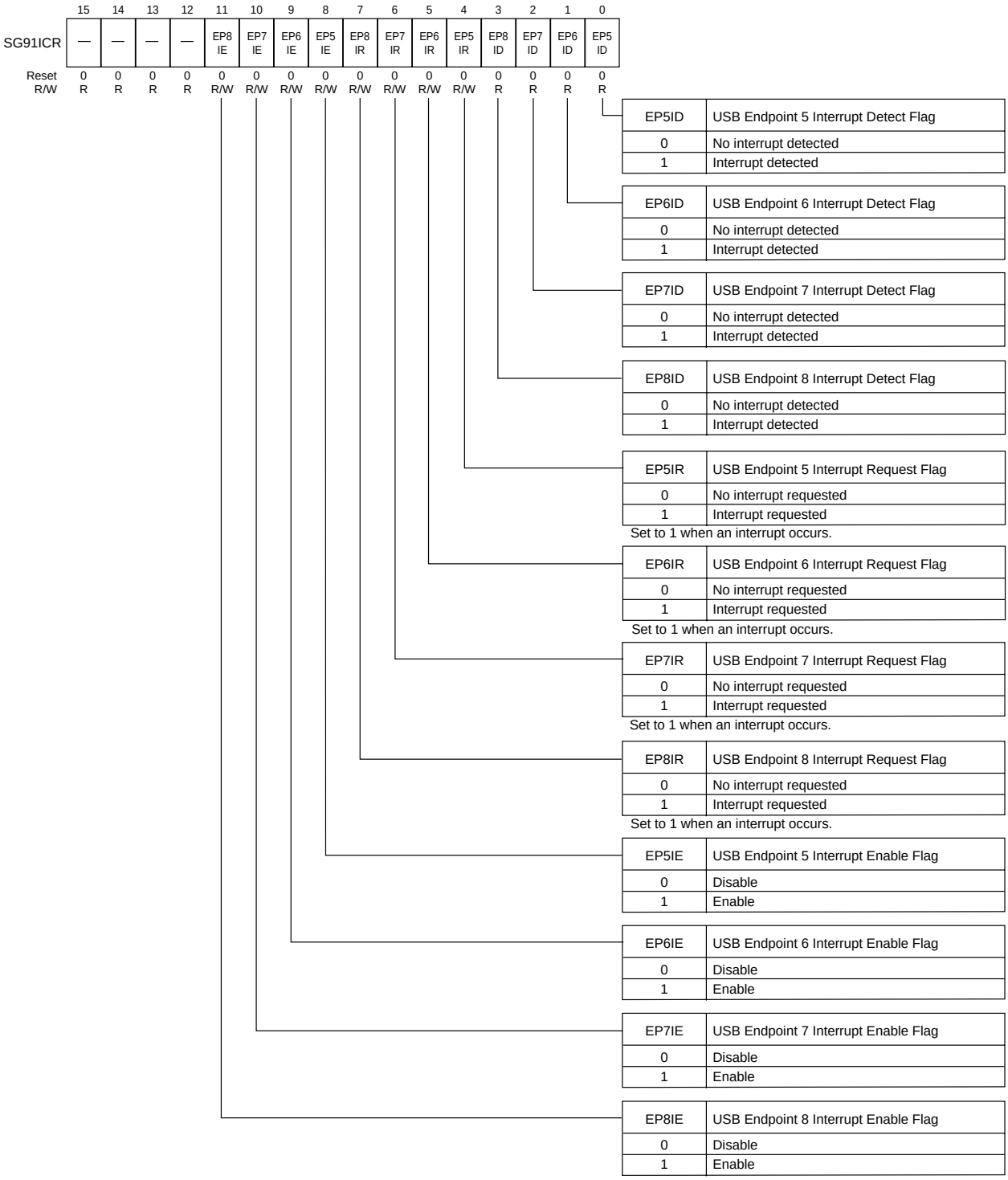

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SG91ICR — — — — EP8 IE EP7 IE EP6 IE EP5 IE EP8 IR EP7 IR EP6 IR EP5 IR EP8 ID EP7 ID EP6 ID EP5 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R/W R R R R R/W R/W R/W R/W R/W R/W R/W R/W R R R R
EP5ID USB Endpoint 5 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP6ID USB Endpoint 6 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP7ID USB Endpoint 7 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP8ID USB Endpoint 8 Interrupt Detect Flag
0 No interrupt detected
1 Interrupt detected
EP5IR USB Endpoint 5 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP6IR USB Endpoint 6 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP7IR USB Endpoint 7 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP8IR USB Endpoint 8 Interrupt Request Flag
0 No interrupt requested
1 Interrupt requested
Set to 1 when an interrupt occurs.
EP5IE USB Endpoint 5 Interrupt Enable Flag
0 Disable
1 Enable
EP6IE USB Endpoint 6 Interrupt Enable Flag
0 Disable
1 Enable
EP7IE USB Endpoint 7 Interrupt Enable Flag
0 Disable
1 Enable
EP8IE USB Endpoint 8 Interrupt Enable Flag
0 Disable
1 Enable

Figure 4-2-22 Subgroup Maskable Interrupt Control Register 91 (SG91ICR: x'E0061E', R/W)

Chapter 5 Timers

5-1 Summary of 8-bit Timer

The MN102H74G/74F/74D/F74G contains ten reload timers which can be used as interval timers, event counters, clock output (timer underflow/2), base clock for serial interface or start timing of A/D conversion.

5-1-1 List of 8-bit Timer Function

The following table shows functions of 8-bit timers.

Table 5-1-1 List of 8-bit Timer Functions

Function \ Timer	8-bit Timer					
	Timer 0	Timer 1	Timer 2	Timer 3	Timer 4	Timer 5
Interrupt Request	Group 8 (G8ICR) TM0IR	Group 9 (G9ICR) TM1IR	Subgroup 5 (SG50ICR) TM2IR	Subgroup 5 (SG50ICR) TM3IR	Subgroup 6 (SG60ICR) TM4IR	Subgroup 6 (SG60ICR) TM5IR
Interrupt Source	Timer 0 Underflow	Timer 1 Underflow	Timer 2 Underflow	Timer 3 Underflow	Timer 4 Underflow	Timer 5 Underflow
Clock Source	TM0IO pin Prescaler 0 SYSCLK XI	TM1IO pin Prescaler 0 SYSCLK	SYSCLK SYSCLK/8 SYSCLK/32 TM2IO pin Timer 3 Undeflow Timer 4 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM3IO pin Timer 2 Underflow Timer 4 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM4IO pin Timer 2 Underflow Timer 3 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM5IO pin Timer 2 Underflow Timer 3 Underflow Timer 4 Underflow
Counter	Down Counter	Down Counter	Down Counter	Down Counter	Down Counter	Down Counter
Interval Timer	✓	✓	✓	✓	✓	✓
Event Counter	✓	✓	✓	✓	✓	✓
Timer Output	✓	✓	✓	✓	✓	✓
Clock Source for Serial Interface	-	-	-	-	-	-
A/D Conversion Trigger	-	-	✓	-	-	-
Cascade		✓		✓	✓	✓

Function \ Timer	8-bit Timer			
	Timer 6	Timer 7	Timer 8	Timer 9
Interrupt Request	Subgroup 6 (SG61ICR) TM6IR	Subgroup 6 (SG61ICR) TM7IR	Subgroup 6 (SG61ICR) TM8IR	Subgroup 6 (SG61ICR) TM9IR
Interrupt Source	Timer 6 Underflow	Timer 7 Underflow	Timer 8 Underflow	Timer 9 Underflow
Clock Source	SYSCLK SYSCLK/8 SYSCLK/32 TM6IO pin Timer 7 Underflow Timer 8 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM7IO pin Timer 6 Underflow Timer 8 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM8IO pin Timer 6 Underflow Timer 7 Underflow	SYSCLK SYSCLK/8 SYSCLK/32 TM9IO pin Timer 6 Underflow Timer 7 Underflow Timer 8 Underflow
Counter	Down Counter	Down Counter	Down Counter	Down Counter
Interval Timer	✓	✓	✓	✓
Event Counter	✓	✓	✓	✓
Timer Output	✓	✓	✓	✓
Clock Source for Serial Interface	✓	✓	✓	✓
A/D Conversion Trigger	-	-	-	-
Cascade		✓	✓	✓

5-1-2 Summary of Timer 0 and Timer 1

■ Timer 0 and Timer 1

Timer 0 and timer 1 are 8-bit down counters which operate based on the set value of the timer base register plus 1. Do not set 0 to the timer base register. An interrupt occurs when timer underflows (the binary counter value x'00' is changed to the new 8-bit value). Timer 0 and timer 1 are used as interval timers, event counters or clock output.

> Timer 0 and timer 1 can cascade. For example, cascading timer 0 and timer 1 forms a 16-bit timer.

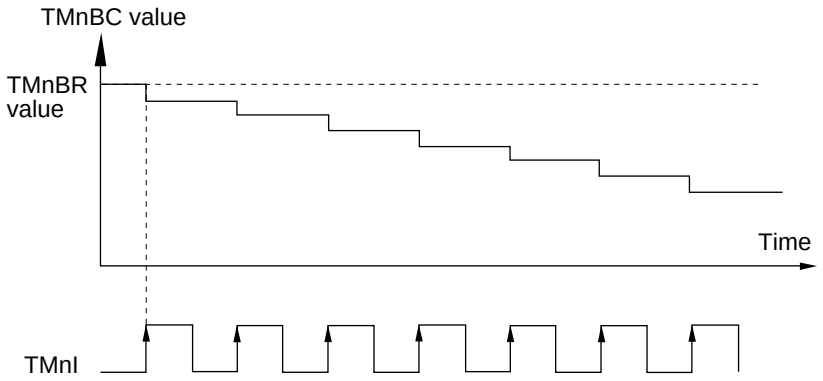


Figure 5-1-1 Event Counter Timing (Timer 0 and Timer 1)

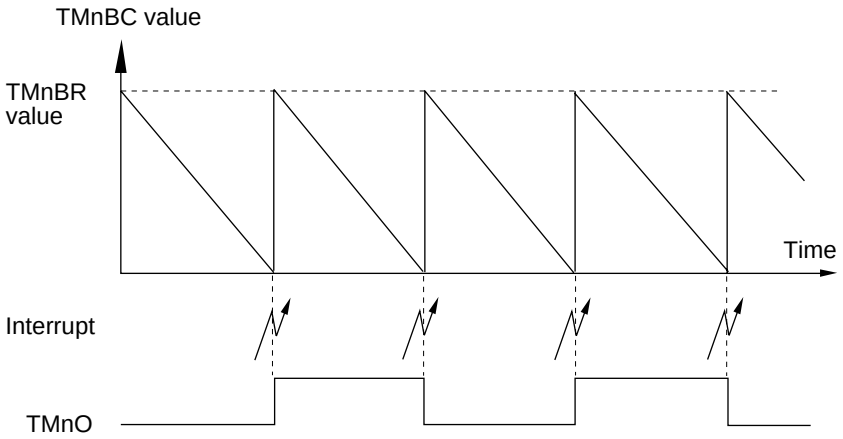


Figure 5-1-2 Timer Ouput, Interval Timer Timing (Timer 0 and Timer 1)

■ Prescaler 0

Prescaler 0 is a 8-bit prescaler which supplies a count clock to timer 0 and timer 1 as the divisor circuit of system clock. Prescaler 0 provides the low-frequency and synchronizes timers easily.

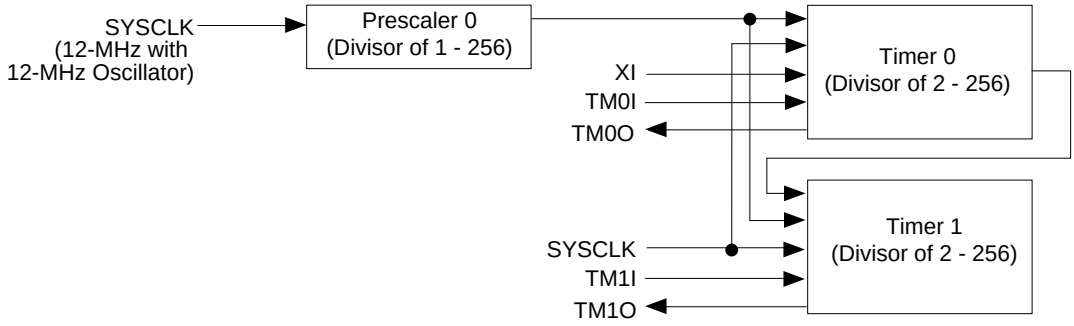


Figure 5-1-3 Relationship between Prescaler, Timer 0 and Timer 1

The following figure shows the timer configuration. Combining 16-bit timer and prescaler forms various interval timers.

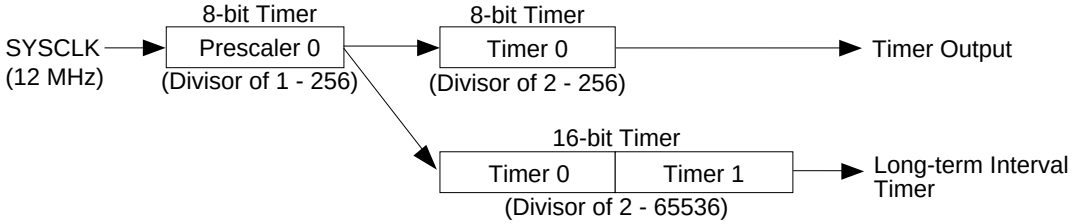


Figure 5-1-4 System Configuration

For cascade connection of timer 1, cascade output of timer 0 is input. Therefore, it operates not as 8 bit dividing which is divided by 8, but as 16 bit dividing (16 bit counter). Also, output of prescaler 0 can be input to timer 0 to 1. At NORMAL mode and HALT mode, SYSCLK becomes the signal (12 MHz at 12 MHz oscillation) which divides BOSC by 2. At STOP mode, SYSCLK stops. SYSCLK can be output to the external SYSCLK pin.

5-1-3 Timer 0 and Timer 1 Block Diagram

This section shows the block diagram of timer 0 and timer 1.

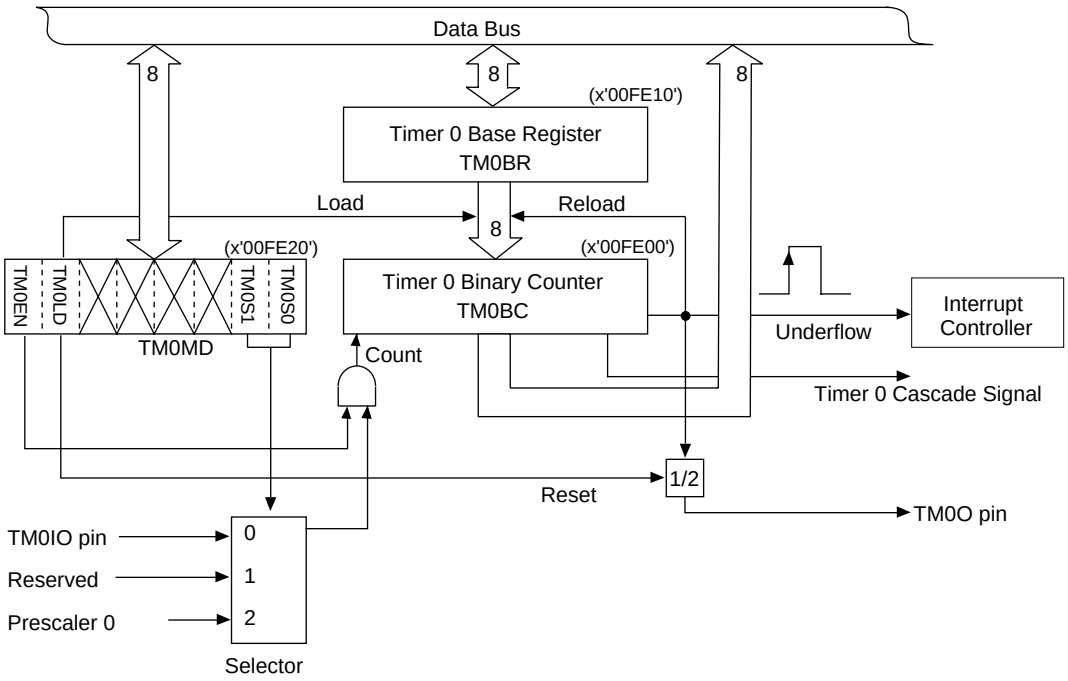


Figure 5-1-5 Timer 0 Block Diagram

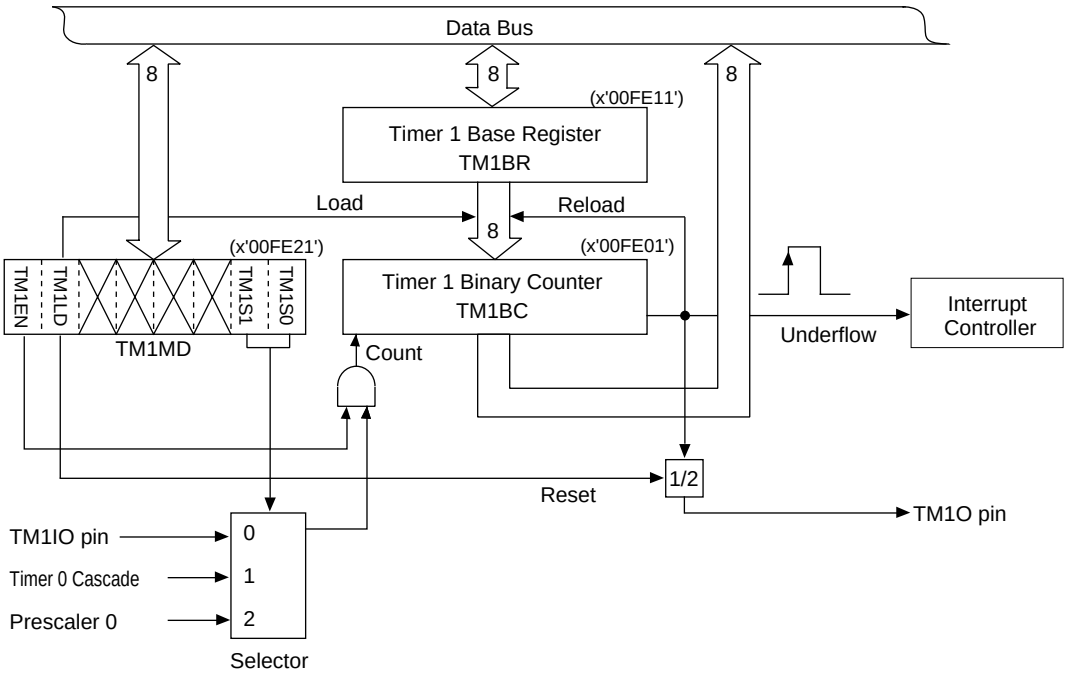


Figure 5-1-6 Timer 1 Block Diagram

5-1-4 Prescaler 0 Block Diagram

This section shows the block diagram of prescaler 0.

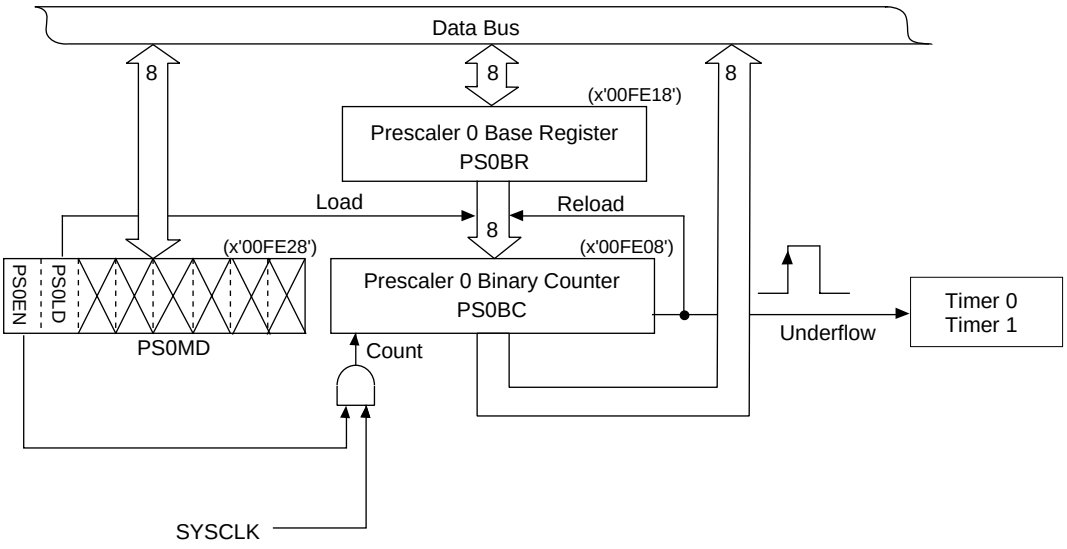


Figure 5-1-7 Prescaler 0 Block Diagram

5-1-5 Timer 2 to Timer 9 Block Diagram

The figure 5-1-8 shows the block diagram of timer 2 to timer 9. The figure 5-1-9 and the figure 5-1-10 show the 8-bit timer connections.

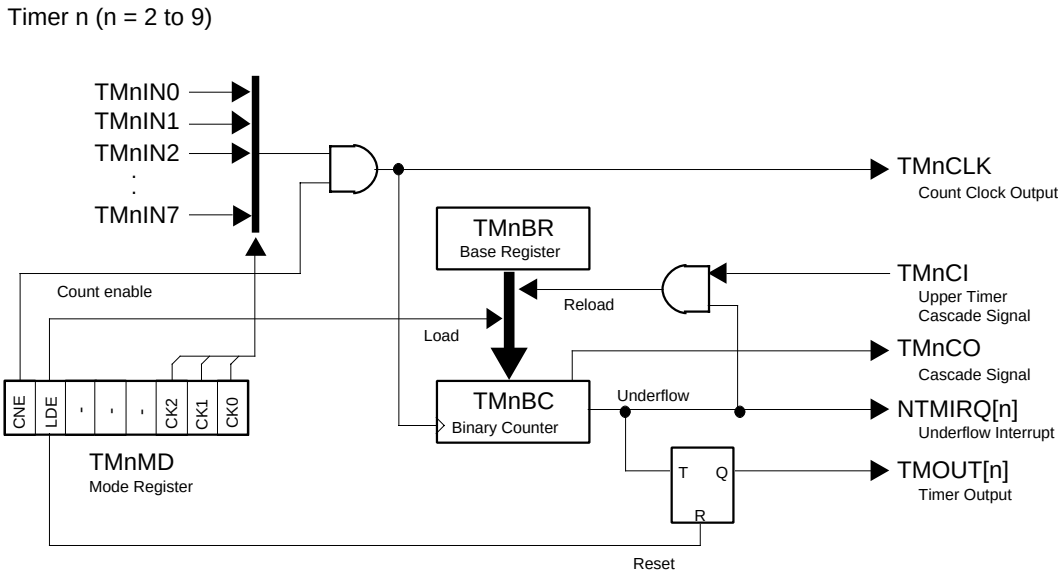


Figure 5-1-8 8-bit Timer Block Diagram (Timer 2 to Timer 9)

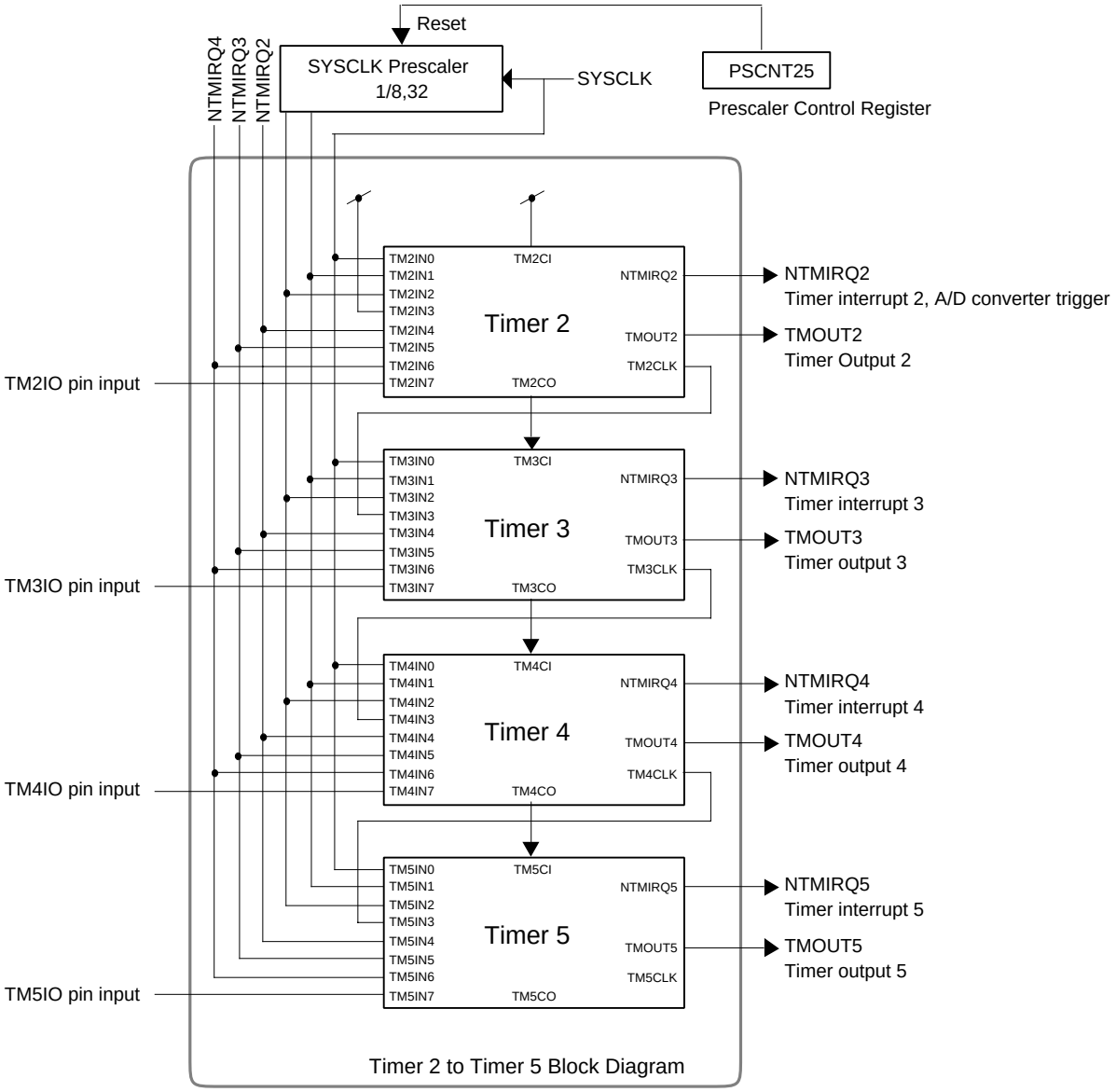


Figure 5-1-9 8-bit Timer Connection (Timer 2 to Timer 5)

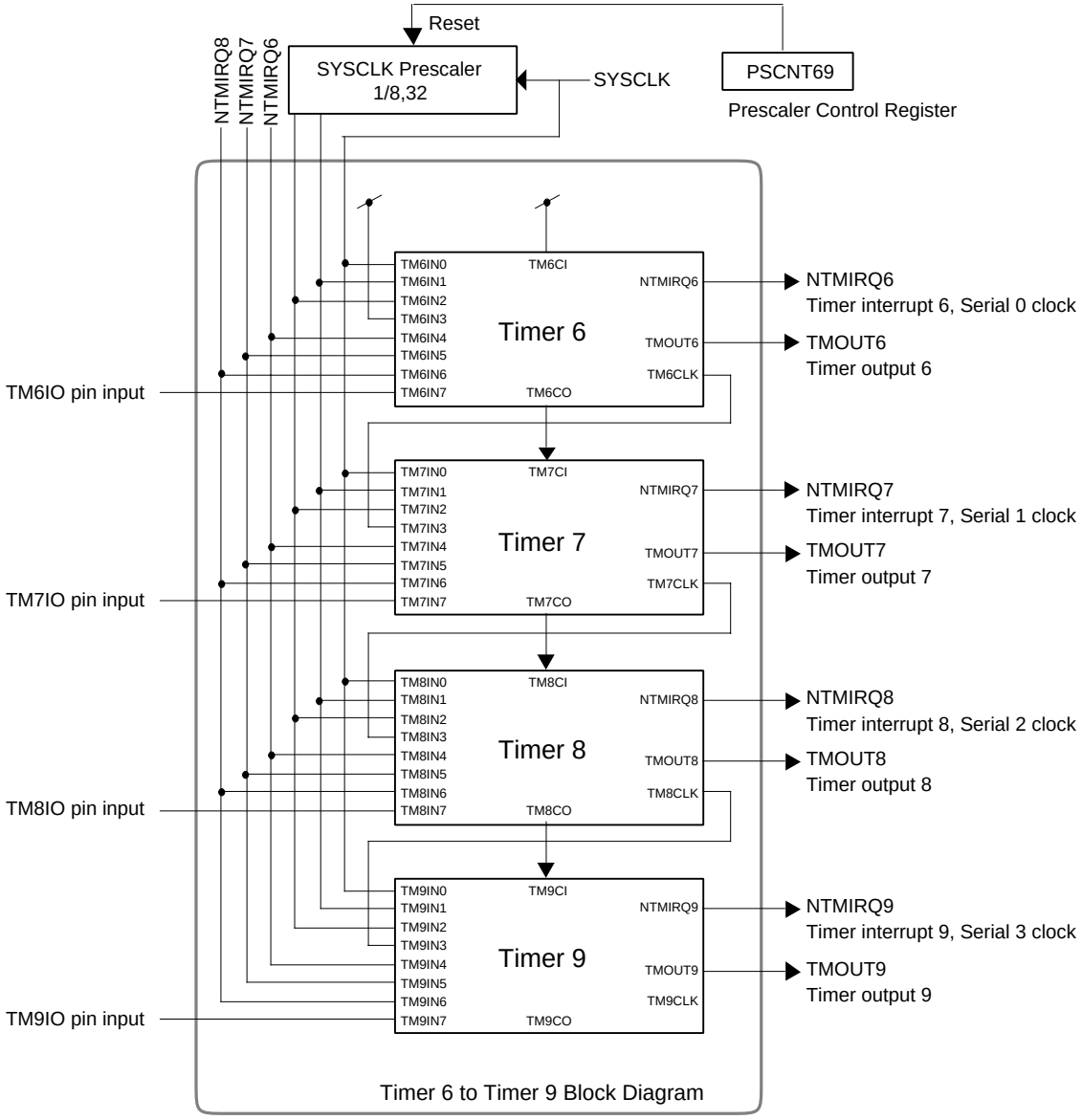


Figure 5-1-10 8-bit Timer Connection (Timer 6 to Timer 9)

5-2 8-bit Timer Control Registers

5-2-1 List of Timer 0, Timer 1 and Prescaler 0 Control Registers

The following table lists the timer 0, timer 1 and prescaler 0 control registers.

Table 5-2-1 List of Timer 0, Timer 1 and Prescaler 0 Control Registers

Register		Address	R/W	Function
Timer 0	TM0MD	x'00FE20'	R/W	Timer 0 Mode Register
	TM0BC	x'00FE00'	R	Timer 0 Binary Counter
	TM0BR	x'00FE10'	R/W	Timer 0 Base Register
Timer 1	TM1MD	x'00FE21'	R/W	Timer 1 Mode Register
	TM1BC	x'00FE01'	R	Timer 1 Binary Counter
	TM1BR	x'00FE11'	R/W	Timer 1 Base Register
Prescaler 0	PS0MD	x'00FE28'	R/W	Prescaler 0 Mode Register
	PS0BC	x'00FE08'	R	Prescaler 0 Binary Counter
	PS0BR	x'00FE18'	R/W	Prescaler 0 Base Register

5-2-2 Timer 0, Timer 1 and Prescaler 0 Programmable Timer Registers

■ Timer n Base Register (n=0 to 1)

The timer n base register sets the count cycle (2 to 256) of timer n. The timer n counts the set value plus 1. The valid range for the timer n base register is 1 to 255.

	7	6	5	4	3	2	1	0
	TM0 BR7	TM0 BR6	TM0 BR5	TM0 BR4	TM0 BR3	TM0 BR2	TM0 BR1	TM0 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-1 Timer 0 Base Register (TM0BR: x'00FE10')

	7	6	5	4	3	2	1	0
	TM1 BR7	TM1 BR6	TM1 BR5	TM1 BR4	TM1 BR3	TM1 BR2	TM1 BR1	TM1 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-2 Timer 1 Base Register (TM1BR: x'00FE11')

> Setting 0 to the timer 0 base register and the timer 1 base register after activating timers. Please refer to "5-3 8-bit Timer 0, 1 Setup Example" for details.

■ Prescaler 0 Base Register

The prescaler 0 base register sets the count cycle (1 to 256) of prescaler 0. The prescaler 0 counts the set value plus 1. The valid range set to the prescaler 0 base register is 0 to 255.

	7	6	5	4	3	2	1	0
	PS0 BR7	PS0 BR6	PS0 BR5	PS0 BR4	PS0 BR3	PS0 BR2	PS0 BR1	PS0 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-3 Prescaler 0 Base Register (PS0BR: x'00FE18')

> Setting 0 to the timer 0 base register and the timer 1 base register after activating timers. Please refer to "5-3 8-bit Timer 0, 1 Setup Example" for details.

■ Timer n Binary Counter (n=0 to 1)

The timer n binary register reads the counter value. This register is read-only.

	7	6	5	4	3	2	1	0
	TM0 BC7	TM0 BC6	TM0 BC5	TM0 BC4	TM0 BC3	TM0 BC2	TM0 BC1	TM0 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-4 Timer 0 Binary Counter (TM0BC: x'00FE00')

	7	6	5	4	3	2	1	0
	TM1 BC7	TM1 BC6	TM1 BC5	TM1 BC4	TM1 BC3	TM1 BC2	TM1 BC1	TM1 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-5 Timer 1 Binary Counter (TM1BC: x'00FE01')

■ Prescaler 0 Binary Counter

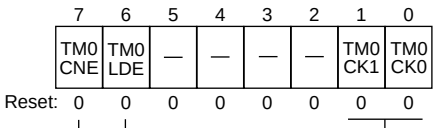
The prescaler 0 binary register reads the counter value. This register is read-only.

	7	6	5	4	3	2	1	0
	PS0 BC7	PS0 BC6	PS0 BC5	PS0 BC4	PS0 BC3	PS0 BC2	PS0 BC1	PS0 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-6 Prescaler 0 Binary Counter (PS0BC: x'00FE08')

5-2-3 Timer 0, Timer 1 and Prescaler 0 Mode Registers

■ Timer 0 Mode Register



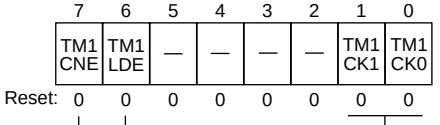
TM0 CK1	TM0 CK0	Count Clock Select
0	0	TM0IO pin input
0	1	XI
1	0	Prescaler 0 clock
1	1	SYSCLK

TM0LDE	Timer 0 Base Register Setup
0	Normal Operation
1	Initialization Load the TM0BR value to the TM0BC Reset timer pulse output 0 to low level

TM0CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-7 Timer 0 Mode Register (TM0MD: x'00FE20', R/W)

■ Timer 1 Mode Register



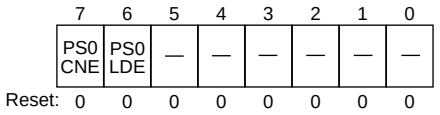
TM1 CK1	TM1 CK0	Count Clock Select
0	0	TM1IO pin input
0	1	Timer 0 cascade
1	0	Prescaler 0 clock
1	1	SYSCLK

TM1LDE	Timer 1 Base Register Setup
0	Normal Operation
1	Initialization Load the TM1BR value to the TM1BC Reset timer pulse output 1 to low level

TM1CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-8 Timer 1 Mode Register (TM1MD: x'00FE21', R/W)

■ Prescaler 0 Mode Register



PS0LDE	Prescaler 0 Base Register Setup
0	Normal Operation
1	Load the PS0BR value to the PS0BC

PS0CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-9 Prescaler 0 Mode Register (PS0MD: x'00FE28', R/W)

5-2-4 List of Timer 2 to Timer 9 Control Registers

The following table lists the timer 2 to timer 9 control registers.

Table 5-2-2 List of Timer 2 to Timer 9 Control Registers

Register		Address	R/W	Function
Timer 2	TM2MD	x'E00030'	R/W	Timer Mode Register 2
	TM2BR	x'E00020'	R/W	Timer Base Register 2
	TM2BC	x'E00010'	R	Timer Binary Counter 2
Timer 3	TM3MD	x'E00031'	R/W	Timer Mode Register 3
	TM3BR	x'E00021'	R/W	Timer Base Register 3
	TM3BC	x'E00011'	R	Timer Binary Counter 3
Timer 4	TM4MD	x'E00032'	R/W	Timer Mode Register 4
	TM4BR	x'E00022'	R/W	Timer Base Register 4
	TM4BC	x'E00012'	R	Timer Binary Counter 4
Timer 5	TM5MD	x'E00033'	R/W	Timer Mode Register 5
	TM5BR	x'E00023'	R/W	Timer Base Register 5
	TM5BC	x'E00013'	R	Timer Binary Counter 5
Timer 6	TM6MD	x'E00034'	R/W	Timer Mode Register 6
	TM6BR	x'E00024'	R/W	Timer Base Register 6
	TM6BC	x'E00014'	R	Timer Binary Counter 6
Timer 7	TM7MD	x'E00035'	R/W	Timer Mode Register 7
	TM7BR	x'E00025'	R/W	Timer Base Register 7
	TM7BC	x'E00015'	R	Timer Binary Counter 7
Timer 8	TM8MD	x'E00036'	R/W	Timer Mode Register 8
	TM8BR	x'E00026'	R/W	Timer Base Register 8
	TM8BC	x'E00016'	R	Timer Binary Counter 8
Timer 9	TM9MD	x'E00037'	R/W	Timer Mode Register 9
	TM9BR	x'E00027'	R/W	Timer Base Register 9
	TM9BC	x'E00017'	R	Timer Binary Counter 9
Prescaler 25	PSCNT25	x'E00000'	R/W	Prescaler 25 Control Register
Prescaler 69	PSCNT69	x'E00004'	R/W	Prescaler 69 Control Regiaser

5-2-5 Timer 2 to Timer 9 Programmable Timer Registers

■ Timer n Base Register (n=2 to 9)

The timer n base register sets the initial value of timer n binary counter and the underflow cycle of timer n. The valid range for the TMnBR is 1 to 255 (at using underflow signal). The TMnBR value is loaded to the TMnBC register under the following two conditions.

1. The TMnLDE flag of the TMnMD register is 1.
2. The timer underflow occurs.

A timer n underflow interrupt occurs whenever the TMnBC counter counts the TMnBR setup plus 1 times.

	7	6	5	4	3	2	1	0
	TM2 BR7	TM2 BR6	TM2 BR5	TM2 BR4	TM2 BR3	TM2 BR2	TM2 BR1	TM2 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-10 Timer 2 Base Register (TM2BR: x'E00020')

	7	6	5	4	3	2	1	0
	TM3 BR7	TM3 BR6	TM3 BR5	TM3 BR4	TM3 BR3	TM3 BR2	TM3 BR1	TM3 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-11 Timer 3 Base Register (TM3BR: x'E00021')

	7	6	5	4	3	2	1	0
	TM4 BR7	TM4 BR6	TM4 BR5	TM4 BR4	TM4 BR3	TM4 BR2	TM4 BR1	TM4 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-12 Timer 4 Base Register (TM4BR: x'E00022')

	7	6	5	4	3	2	1	0
	TM5 BR7	TM5 BR6	TM5 BR5	TM5 BR4	TM5 BR3	TM5 BR2	TM5 BR1	TM5 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-13 Timer 5 Base Register (TM5BR: x'E00023')

	7	6	5	4	3	2	1	0
	TM6 BR7	TM6 BR6	TM6 BR5	TM6 BR4	TM6 BR3	TM6 BR2	TM6 BR1	TM6 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-14 Timer 6 Base Register (TM6BR: x'E00024')

	7	6	5	4	3	2	1	0
	TM7 BR7	TM7 BR6	TM7 BR5	TM7 BR4	TM7 BR3	TM7 BR2	TM7 BR1	TM7 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-15 Timer 7 Base Register (TM7BR: x'E00025')

	7	6	5	4	3	2	1	0
	TM8 BR7	TM8 BR6	TM8 BR5	TM8 BR4	TM8 BR3	TM8 BR2	TM8 BR1	TM8 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-16 Timer 8 Base Register (TM8BR: x'E00026')

	7	6	5	4	3	2	1	0
	TM9 BR7	TM9 BR6	TM9 BR5	TM9 BR4	TM9 BR3	TM9 BR2	TM9 BR1	TM9 BR0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-17 Timer 9 Base Register (TM9BR: x'E00027')

■ Timer n Binary Counter (n=2 to 9)

The timer n binary counter is down counting and reads the counter value. When the TMnBC counts the TMnBR setup value plus 1 times, a timer n underflow interrupt occurs.

	7	6	5	4	3	2	1	0
	TM2 BC7	TM2 BC6	TM2 BC5	TM2 BC4	TM2 BC3	TM2 BC2	TM2 BC1	TM2 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-18 Timer 2 Binary Counter (TM2BC: x'E00010')

	7	6	5	4	3	2	1	0
	TM3 BC7	TM3 BC6	TM3 BC5	TM3 BC4	TM3 BC3	TM3 BC2	TM3 BC1	TM3 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-19 Timer 3 Binary Counter (TM3BC: x'E00011')

	7	6	5	4	3	2	1	0
	TM4 BC7	TM4 BC6	TM4 BC5	TM4 BC4	TM4 BC3	TM4 BC2	TM4 BC1	TM4 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-20 Timer 4 Binary Counter (TM4BC: x'E00012')

	7	6	5	4	3	2	1	0
	TM5 BC7	TM5 BC6	TM5 BC5	TM5 BC4	TM5 BC3	TM5 BC2	TM5 BC1	TM5 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-21 Timer 5 Binary Counter (TM5BC: x'E00013')

	7	6	5	4	3	2	1	0
	TM6 BC7	TM6 BC6	TM6 BC5	TM6 BC4	TM6 BC3	TM6 BC2	TM6 BC1	TM6 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-22 Timer 6 Binary Counter (TM6BC: x'E00014')

	7	6	5	4	3	2	1	0
	TM7 BC7	TM7 BC6	TM7 BC5	TM7 BC4	TM7 BC3	TM7 BC2	TM7 BC1	TM7 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-23 Timer 7 Binary Counter (TM7BC: x'E00015')

	7	6	5	4	3	2	1	0
	TM8 BC7	TM8 BC6	TM8 BC5	TM8 BC4	TM8 BC3	TM8 BC2	TM8 BC1	TM8 BC0
Reset	0	0	0	0	0	0	0	0

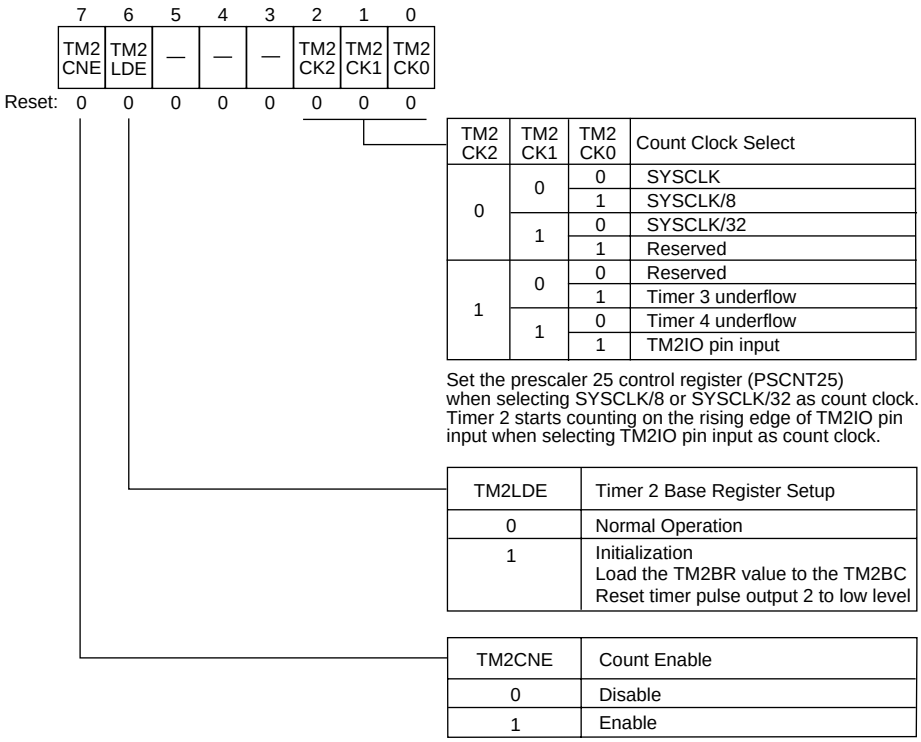
Figure 5-2-24 Timer 8 Binary Counter (TM8BC: x'E00016')

	7	6	5	4	3	2	1	0
	TM9 BC7	TM9 BC6	TM9 BC5	TM9 BC4	TM9 BC3	TM9 BC2	TM9 BC1	TM9 BC0
Reset	0	0	0	0	0	0	0	0

Figure 5-2-25 Timer 9 Binary Counter (TM9BC: x'E00017')

5-2-6 Timer 2 to Timer 9 Mode Registers

■ Timer 2 Mode Register



TM2 CK2	TM2 CK1	TM2 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Reserved
1	0	0	Reserved
		1	Timer 3 underflow
	1	0	Timer 4 underflow
		1	TM2IO pin input

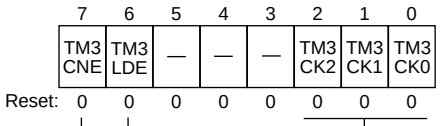
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 2 starts counting on the rising edge of TM2IO pin input when selecting TM2IO pin input as count clock.

TM2LDE	Timer 2 Base Register Setup
0	Normal Operation
1	Initialization Load the TM2BR value to the TM2BC Reset timer pulse output 2 to low level

TM2CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-26 Timer 2 Mode Register (TM2MD: x'E00030', R/W)

■ Timer 3 Mode Register



TM3 CK2	TM3 CK1	TM3 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 2 cascade
1	0	0	Timer 2 underflow
		1	Reserved
	1	0	Timer 4 underflow
		1	TM3IO pin input

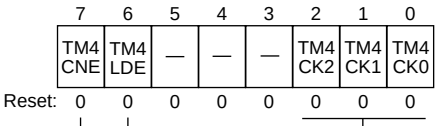
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 3 starts counting on the rising edge of TM3IO pin input when selecting TM3IO pin input as count clock.

TM3LDE	Timer 3 Base Register Setup
0	Normal Operation
1	Initialization Load the TM3BR value to the TM3BC Reset timer pulse output 3 to low level

TM3CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-27 Timer 3 Mode Register (TM3MD: x'E00031', R/W)

■ Timer 4 Mode Register



TM4 CK2	TM4 CK1	TM4 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 3 cascade
1	0	0	Timer 2 underflow
		1	Timer 3 underflow
	1	0	Reserved
		1	TM4IO pin input

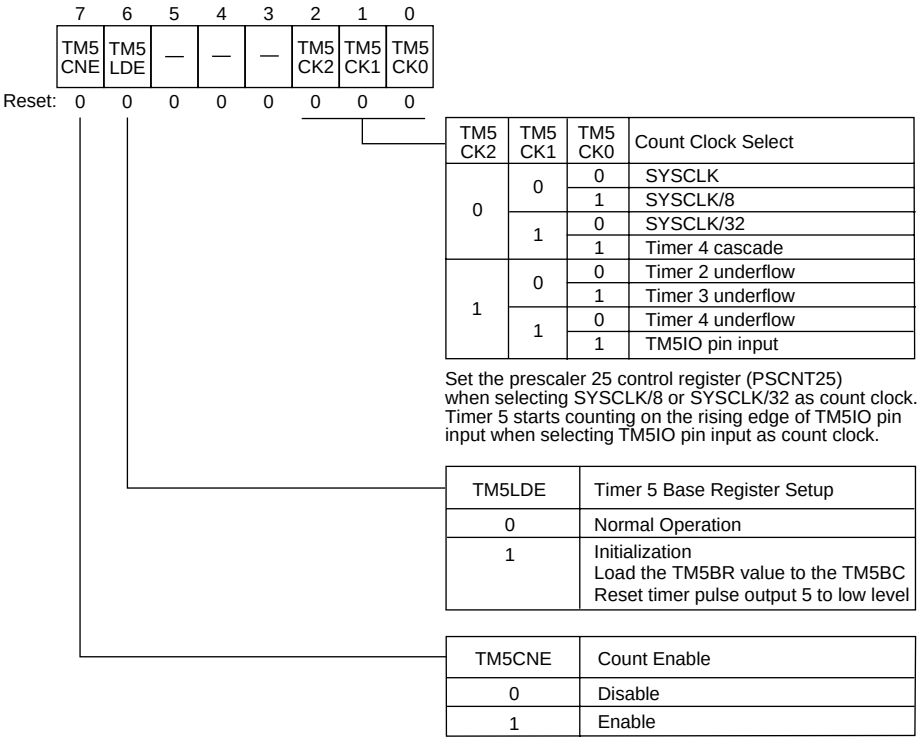
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock.
Timer 4 starts counting on the rising edge of TM4IO pin input when selecting TM4IO pin input as count clock.

TM4LDE	Timer 4 Base Register Setup
0	Normal Operation
1	Initialization Load the TM4BR value to the TM4BC Reset timer pulse output 4 to low level

TM4CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-28 Timer 4 Mode Register (TM4MD: x'E00032', R/W)

■ Timer 5 Mode Register



TM5 CK2	TM5 CK1	TM5 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 4 cascade
1	0	0	Timer 2 underflow
		1	Timer 3 underflow
	1	0	Timer 4 underflow
		1	TM5IO pin input

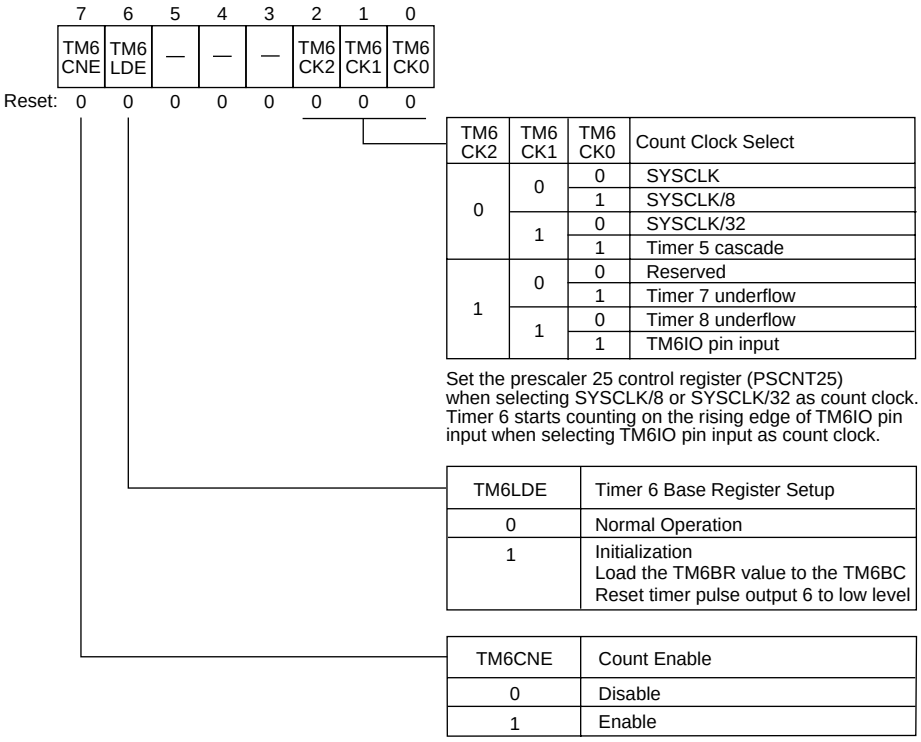
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 5 starts counting on the rising edge of TM5IO pin input when selecting TM5IO pin input as count clock.

TM5LDE	Timer 5 Base Register Setup
0	Normal Operation
1	Initialization Load the TM5BR value to the TM5BC Reset timer pulse output 5 to low level

TM5CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-29 Timer 5 Mode Register (TM5MD: x'E00033', R/W)

■ Timer 6 Mode Register


7 6 5 4 3 2 1 0
TM6 CNE
TM6 LDE
—
—
—
TM6 CK2
TM6 CK1
TM6 CK0
Reset: 0 0 0 0 0 0 0 0
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 6 starts counting on the rising edge of TM6IO pin input when selecting TM6IO pin input as count clock.

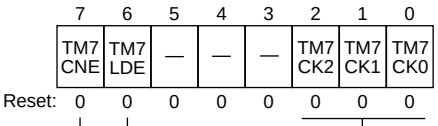
TM6 CK2	TM6 CK1	TM6 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 5 cascade
1	0	0	Reserved
		1	Timer 7 underflow
	1	0	Timer 8 underflow
		1	TM6IO pin input

TM6LDE	Timer 6 Base Register Setup
0	Normal Operation
1	Initialization Load the TM6BR value to the TM6BC Reset timer pulse output 6 to low level

TM6CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-30 Timer 6 Mode Register (TM6MD: x'E00034', R/W)

■ Timer 7 Mode Register



TM7 CK2	TM7 CK1	TM7 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 6 cascade
1	0	0	Timer 6 underflow
		1	Reserved
	1	0	Timer 8 underflow
		1	TM7IO pin input

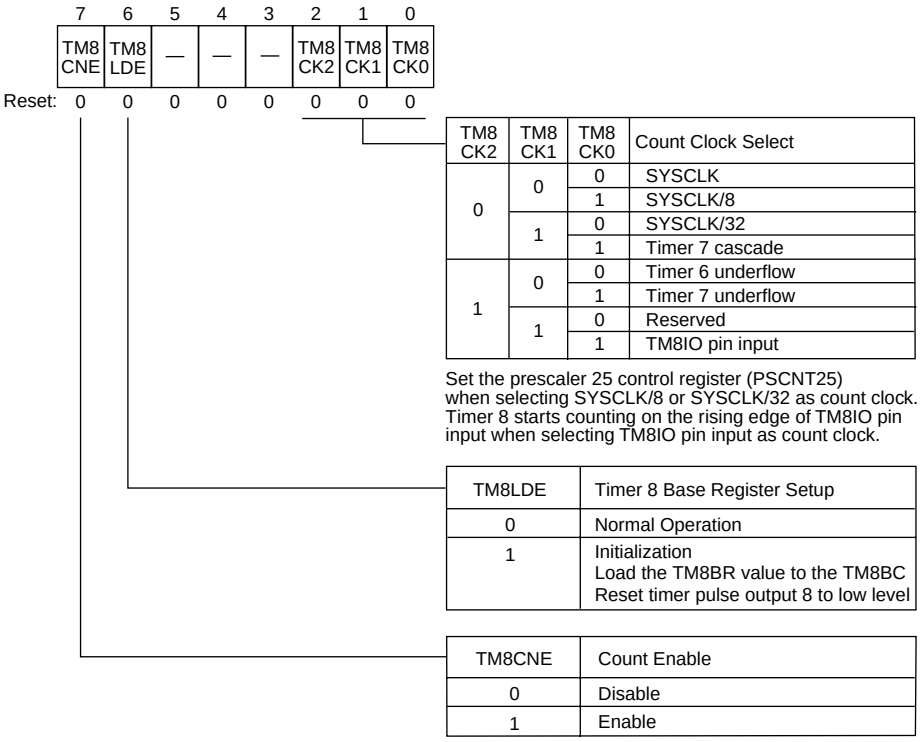
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 7 starts counting on the rising edge of TM7IO pin input when selecting TM7IO pin input as count clock.

TM7LDE	Timer 7 Base Register Setup
0	Normal Operation
1	Initialization Load the TM7BR value to the TM7BC Reset timer pulse output 7 to low level

TM7CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-31 Timer 7 Mode Register (TM7MD: x'E00035', R/W)

■ Timer 8 Mode Register



TM8 CK2	TM8 CK1	TM8 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 7 cascade
1	0	0	Timer 6 underflow
		1	Timer 7 underflow
	1	0	Reserved
		1	TM8IO pin input

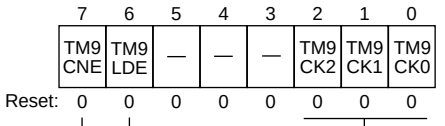
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 8 starts counting on the rising edge of TM8IO pin input when selecting TM8IO pin input as count clock.

TM8LDE	Timer 8 Base Register Setup
0	Normal Operation
1	Initialization Load the TM8BR value to the TM8BC Reset timer pulse output 8 to low level

TM8CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-32 Timer 8 Mode Register (TM8MD: x'E00036', R/W)

■ Timer 9 Mode Register



TM9 CK2	TM9 CK1	TM9 CK0	Count Clock Select
0	0	0	SYSCLK
		1	SYSCLK/8
	1	0	SYSCLK/32
		1	Timer 8 cascade
1	0	0	Timer 6 underflow
		1	Timer 7 underflow
	1	0	Timer 8 underflow
		1	TM9IO pin input

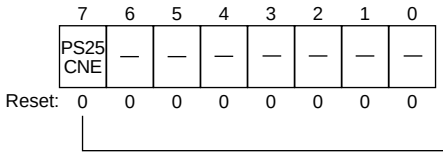
Set the prescaler 25 control register (PSCNT25) when selecting SYSCLK/8 or SYSCLK/32 as count clock. Timer 9 starts counting on the rising edge of TM9IO pin input when selecting TM9IO pin input as count clock.

TM9LDE	Timer 9 Base Register Setup
0	Normal Operation
1	Initialization Load the TM9BR value to the TM9BC Reset timer pulse output 9 to low level

TM9CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-33 Timer 9 Mode Register (TM9MD: x'E00037', R/W)

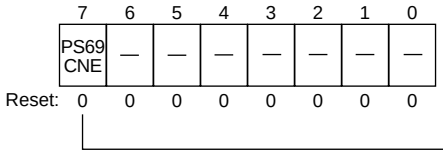
■ Prescaler 25 Control Register



PS25CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-34 Prescaler 25 Control Register (PSCNT25: x'E00000', R/W)

■ Prescaler 69 Control Register



PS69CNE	Count Enable
0	Disable
1	Enable

Figure 5-2-35 Prescaler 69 Control Register (PSCNT69: x'E00004', R/W)

5-3 Timer 0 and Timer 1 Setup Example

5-3-1 Event Counter Using Timer 0 or Timer 1

In this example, timer 0 generates a timer underflow interrupt on the fourth rising edge of TM0IO input.

(1) Set the interrupt enable flag (IE) of the processor status word (PSW) to 1.

(2) Verify that counting stops by the timer 0 mode register (TM0MD).

TM0MD: x'00FE20'

	7	6	5	4	3	2	1	0
	TM0 EN	TM0 LD	-	-	-	-	TM0 CK1	TM0 CK0
Reset	0	0	-	-	-	-	0	0

Step (2) is unnecessary immediately after a reset.

Verification is completed by reading, modifying and writing.

```
movb  (TMnMD), d0
and   x'7F', d0
movb  d0, (TMnMD)
```

(3) Enable interrupts. At this point, clear all existing interrupt requests. To do this, set IRQ8LV[2:0] flags of the G8ICR register to the interrupt level, TM0IR to 0 and TM0IE to 1. In the following case, write x'4100' to the G8ICR register. Thereafter, an interrupt occurs when timer 0 underflows.

G8ICR: x'00FC50'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IRQ8 LV2	IRQ8 LV1	IRQ8 LV0	-	-	TM0 IE	NRQ14 IE	-	-	TM0 IR	NRQ14 IR	-	-	TM0 ID	NRQ14 ID
-	1	0	0	-	-	0	1	0	0	0	0	-	-	0	0

(4) Set the divisor of timer 0. Set the timer 0 base register (TM0BR) to 3 since the divisor is TM0IO input divided by 4. (The valid range for TM0BR is 1 to 255.)

TM0BR: x'00FE10'

7	6	5	4	3	2	1	0
TM0 BR7	TM0 BR6	TM0 BR5	TM0 BR4	TM0 BR3	TM0 BR2	TM0 BR1	TM0 BR0
0	0	0	0	0	0	1	1

(5) Set the TM0LD flag and the TM0EN flag of the TM0MD register to 1 and 0 respectively. At the same time, select the clock source (set the TM0CK[1:0] flags to 00).

> Selecting clock source while controlling count operation will corrupt the value in the binary counter.

(6) Set both TM0LD and TM0EN of the TM0MD register to 0. If this setup is omitted, the timer 0 binary counter does not operate the first count.

(7) Set TM0LD and TM0EN to 0 and 1 respectively. This starts the timer. Counting begins at the next start of the next cycle.

When the TM0BC counter reaches 0 and loads the value 3 of the TM0BR register at the next count, a timer 0 underflow interrupt occurs.

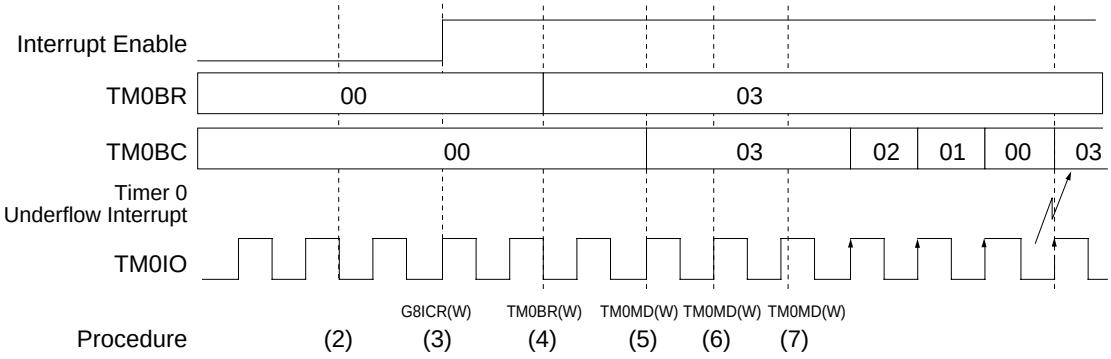


Figure 5-3-1 Event Counter Timing

5-3-2 Clock Output Using Timer 0 or Timer 1

In this example, timer 1 and prescaler 0 output 12 cycles (SYSCLK/6).

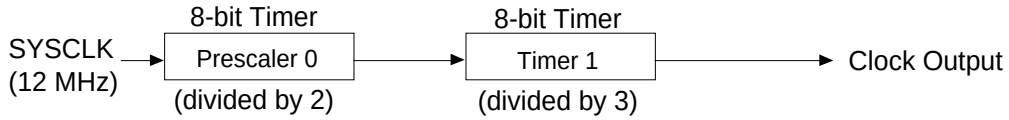


Figure 5-3-2 Clock Output Configuration Example

■ Prescaler 0 Setup

(1) Verify that counting stops by the prescaler 0 mode register (PS0MD).

PS0MD: x'00FE28'

	7	6	5	4	3	2	1	0
	PS0 EN	PS0 LD	-	-	-	-	-	-
Reset	0	0						

Step (1) is unnecessary immediately after a reset.

Verification is completed by read, modify and write.

```
movb  (TMnMD), d0
and   x'7F', d0
movb  d0, (TMnMD)
```

(2) Set the divisor of prescaler 0. Set the prescaler 0 base register (PS0BR) to 1 since the divisor is SYSCLK/2. (The valid range for PS0BR is 0 to 255.)

PS0BR: x'00FE18'

7	6	5	4	3	2	1	0
PS0 BR7	PS0 BR6	PS0 BR5	PS0 BR4	PS0 BR3	PS0 BR2	PS0 BR1	PS0 BR0
0	0	0	0	0	0	0	1

If the divisor is 1, write the dummy data (for example x'0F') once.

(3) Set PS0LD and PS0EN to 1 and 0 to load the PS0BR value to the PS0BC counter.

PS0MD: x'00FE28'

7	6	5	4	3	2	1	0
PS0 EN	PS0 LD	-	-	-	-	-	-
0	1						

(4) Set both PS0LD and PS0EN to 0. If this setup is omitted, the prescaler 0 binary counter does not operate the first count.

(5) Set PS0LD and PS0EN to 0 and 1 respectively. This starts the prescaler. Counting begins at the next start of the next cycle. When the PS0BC counter reaches 0 and loads the value 1 of the PS0BR register at the next count, a prescaler 0 underflow interrupt occurs.

> If the divisor is 1, set the divisor of 0 to the PS0BR register again after step (5). The divisor is the value set in step (2) during the first count, and becomes 1 from the second count. If the divisor is set to 0 in step (2), the divisor becomes 256 on the first count and becomes 1 from the second count.

■ Timer 1 Setup

(6) Verify that counting stops by the timer 1 mode register (TM1MD).

TM1MD: x'00FE21'

	7	6	5	4	3	2	1	0
	TM1 EN	TM1 LD	-	-	-	-	TM1 CK1	TM1 CK0
Reset	0	0	-	-	-	-	0	0

> Step (1) is unnecessary immediately after a reset.

> Verification is completed by read, modify and write.
> movb (TMnMD), d0
> and x'7F', d0
> movb d0, (TMnMD)

(7) Set the divisor of timer 1. Set the timer 1 base register (TM1BR) to 2 since the divisor is prescaler output/3. (The valid range for TM1BR is 1 to 255.)

TM1BR: x'00FE11'

7	6	5	4	3	2	1	0
TM1 BR7	TM1 BR6	TM1 BR5	TM1 BR4	TM1 BR3	TM1 BR2	TM1 BR1	TM1 BR0
0	0	0	0	0	0	1	0

(8) Set the TM1LD flag and the TM1EN flag of the TM1MD register to 1 and 0 respectively. At the same time, select the clock source (set the TM1CK[1:0] flags to 10).

> **!** Selecting clock source while controlling count operation will corrupt the value in the binary counter.

(9) Set both TM1LD and TM1EN to 0. If this setup is omitted, the timer 1 binary counter does not operate the first count.

(10) Set TM1LD and TM1EN to 0 and 1 respectively. This starts the timer. Counting begins at the next start of the next cycle.

When the TM1BC counter reaches 0 and loads the value 2 of the TM1BR register at the next count, invert the TM1IO output. TM1IO outputs 0 immediately after start. Then TM1IO outputs 1 on the beginning of the next count after the TM1BC is 0. Finally TM1IO outputs 0 on the beginning of the next count after the TM1BC becomes 0 again. This operation generates 12 cycles clock output.

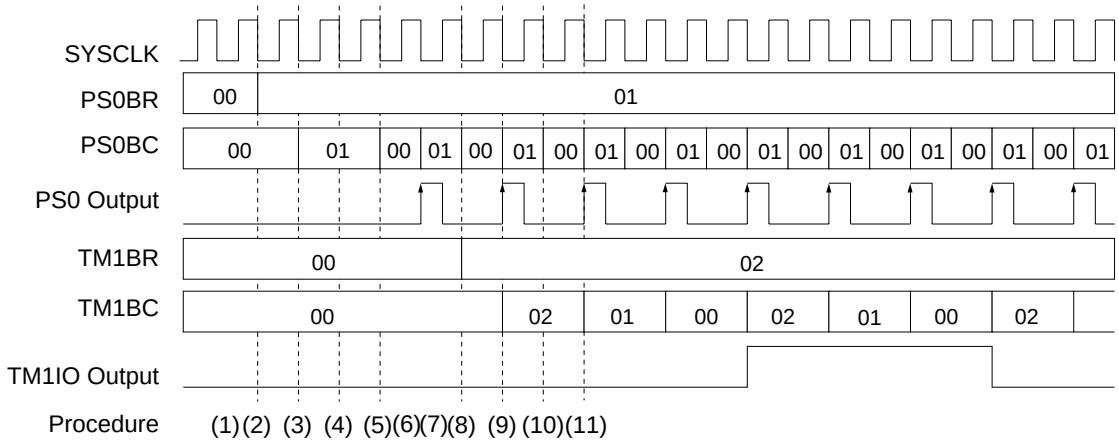


Figure 5-3-3 Clock Output Timing

5-3-3 Interval Timer Using Timer 0 or Timer 1

In this example, timer 0, timer 1 and prescaler 0 generate interrupts at the fixed interval (1 second). (Divide prescaler 0 by 200 and timer 0 , timer 1 by 60000 to divide SYSCLK by 12000000.)

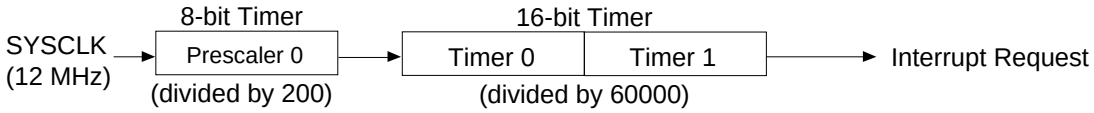


Figure 5-3-4 Interval Timer Configuration Example

(1) Set the interrupt enable flag (IE) of the processor status word (PSW) to 1.

(2) Enable interrupts. At this point, clear all existing interrupt requests. To do this, set IRQ9LV[2:0] flags of the G9ICR register to the interrupt level, TM1IR to 0 and TM1IE to 1. In the following case, write x'4200' to the G9ICR register. Thereafter, an interrupt occurs when timer 0 underflows.

G9ICR: x'00FC52'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IRQ9 LV2	IRQ9 LV1	IRQ9 LV0	-	-	TM1 IE	NIRQ15 IE	-	-	TM1 IR	NIRQ15 IR	-	-	TM1 ID	NIRQ15 ID
-	1	0	0	-	-	1	0	-	-	0	0	-	-	0	0

Step (1) is unnecessary immediately after a reset.

Verification is completed by read, modify and write.
movb (TMnMD), d0
and x'7F', d0
movb d0, (TMnMD)

■ Prescaler 0 Setup

(3) Verify that counting stops by the prescaler 0 mode register (PS0MD).

PS0MD: x'00FE28'

	7	6	5	4	3	2	1	0
	PS0 EN	PS0 LD	-	-	-	-	-	-
Reset	0	0						

(4) Set the divisor of prescaler 0. Set the prescaler 0 base register (PS0BR) to 199 since the divisor is SYSCLK/200. (The valid range for PS0BR is 0 to 255.)

PS0BR: x'00FE18'

7	6	5	4	3	2	1	0
PS0 BR7	PS0 BR6	PS0 BR5	PS0 BR4	PS0 BR3	PS0 BR2	PS0 BR1	PS0 BR0
1	1	0	0	0	1	1	1

If the divisor is 1, write the dummy data (for example x'0F') once.

(5) Set PS0LD and PS0EN to 1 and 0 to load the PS0BR value to the PS0BC counter.

PS0MD: x'00FE28'

7	6	5	4	3	2	1	0
PS0 EN	PS0 LD	-	-	-	-	-	-
0	1						

(6) Set both PS0LD and PS0EN to 0. If this setup is omitted, the prescaler 0 binary counter does not operate the first count.

(7) Set PS0LD and PS0EN to 0 and 1 respectively. This starts the prescaler. Counting begins at the next start of the next cycle. When the PS0BC counter reaches 0 and loads the value 199 of the PS0BR register at the next count, a prescaler 0 underflow interrupt occurs.

If the divisor is 1, set the divisor of 0 to the PS0BR register again after step (7). The divisor is the value set in step (4) during the first count, and becomes 1 from the second count. If the divisor is set to 0 in step (4), the divisor becomes 256 on the first count and becomes 1 from the second count.

■ Timer 0, Timer 1 Setup

(8) Verify that counting stops by the timer 0 mode register (TM0MD) and the timer 1 mode register (TM1MD).

TM0MD: x'00FE20'

	7	6	5	4	3	2	1	0
	TM0 EN	TM0 LD	-	-	-	-	TM0 CK1	TM0 CK0
Reset	0	0	-	-	-	-	0	0

TM1MD: x'00FE21'

	7	6	5	4	3	2	1	0
	TM1 EN	TM1 LD	-	-	-	-	TM1 CK1	TM1 CK0
Reset	0	0	-	-	-	-	0	0

Step (8) is unnecessary immediately after a reset.

Verification is completed by read, modify and write.

```
movb  (TMnMD), d0
and   x'7F', d0
movb  d0, (TMnMD)
```

(9) Set the divisor of the timer. Set the timer 0 base register (TM0BR) to x'5F' and the timer 1 base register (TM1BR) to x'EA' since the divisor is 60000 (x'EA60'). (The valid range for TM0BR and TM1BR is 1 to 255.)

TM0BR: x'00FE10'

	7	6	5	4	3	2	1	0
	TM0 BR7	TM0 BR6	TM0 BR5	TM0 BR4	TM0 BR3	TM0 BR2	TM0 BR1	TM0 BR0
Reset	0	1	0	1	1	1	1	1

TM1BR: x'00FE11'

	7	6	5	4	3	2	1	0
	TM1 BR7	TM1 BR6	TM1 BR5	TM1 BR4	TM1 BR3	TM1 BR2	TM1 BR1	TM1 BR0
Reset	1	1	1	0	1	0	1	0

(10) Set TM0LD and TM0EN to 1 and 0 respectively and TM1LD and TM1EN to 1 and 0 respectively. At the same time, select the clock source (select prescaler 0 for timer 0 and timer 0 cascade for timer 1).

Selecting clock source while controlling count operation will corrupt the value in the binary counter.

(11) Set all TM0LD, TM0EN, TM1LD and TM1EN to 0. If this setup is omitted, the binary counters do not operate the first count.

(12) Set TM0LD and TM0EN to 0 and 1 respectively and TM1LD and TM1EN to 0 and 1 respectively. This starts the timer. Counting begins at the next start of the next cycle.

When the TM0BC counter and the TM1BC counter reach 0, and TM0BC loads the value x'5F' of the TM0BR register and TM1BC load the value x'EA' of the TM1BR register at the next count, a timer 1 underflow interrupt occurs.

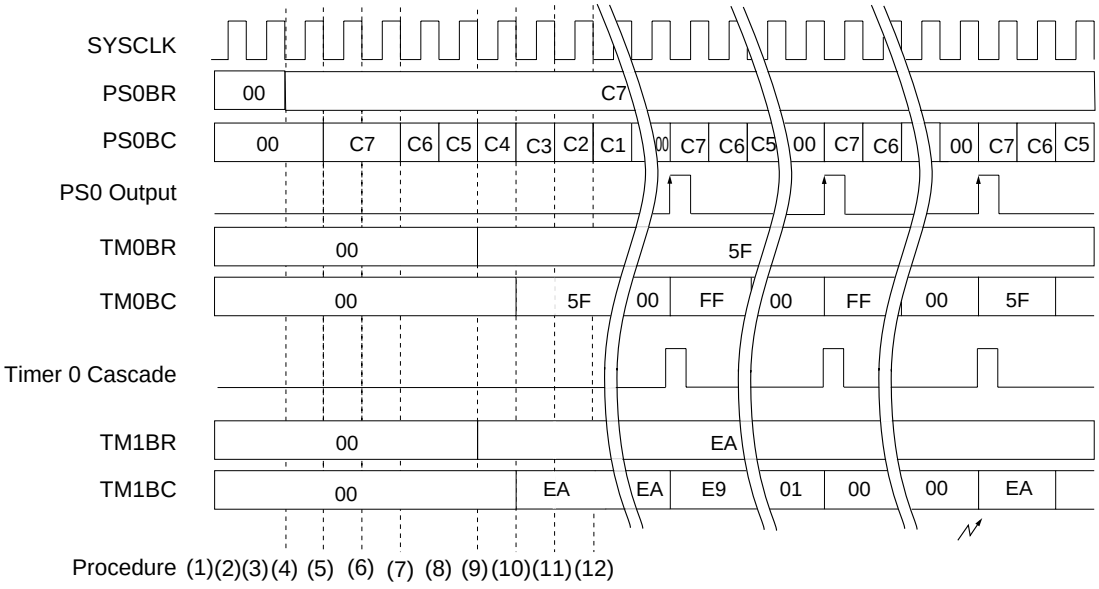


Figure 5-3-5 Interval Timer Timing

5-4 Timer 2 to Timer 9 Setup Example

5-4-1 Interval Timer and Timer Output

This section shows the setup procedures when using a 8-bit timer as an interval timer. The interval timer generates an interrupt at the set cycle.

Please see “5-6 Cascade Connection” when cascading 8-bit timers forms a 16-bit timer, 24-bit timer or 32-bit timer.

■ Timer Start

(1) Set the divisor of timer. Set the divisor to the TMnBR register. The valid range for TMnBR is 1 to 255. The interrupt request cycle should be (TMnBR set value +1) × Clock Source Cycle.

(2) Select Clock Source. Set the TMnCK[2:0] flags of the TMnMD register to the clock source. When selecting SYSCLK/8 or SYSCLK/32 as clock source, enable prescaler operation by setting the PS25CNE flag of the PSCNT25 register or PS69CNE flag of PSCNT69 register to 1.

(3) Initialize the timer by setting the TMnLDE of the TMnMD register to 1. Load the TMnBR value to the TMnBC counter as the initial value and reset the timer output. Set the TMnLDE to 0 and return to the normal operation always after initialization.

Set the TMnLDE flag while the TMnCNE flag is 0.

(4) Enable timer counting. Timer starts counting by setting the TMnCNE flag of the TMnMD register to 1.

When enabling the counting, a timer underflow interrupt occurs at the fixed cycle. Whenever a timer underflow interrupt occurs, invert TMOUT[n] pin output and load the TMnBR value to the TMnBC counter. If changing the TMnBR value during counting, load the new TMnBR value to the TMnBC counter as the initial value when the next timer underflow interrupt occurs and change the interrupt cycle.

■ Timer Stop

(1) Stop the timer counting. Counting stops by setting the TMnCNE flag of the TMnMD register to 0.

(2) Initialize the timer as needed. When setting the TMnLDE of the TMnMD register to 1, load the TMnBR value to the TMnBC counter as the initial value and reset the timer output. When timer stops and TMnLDE is not set to 1, the TMnBC counter and pin output are remain as the status before timer stops. Setting TMnCNE to 1 starts counting from the status immediately before timer stops.

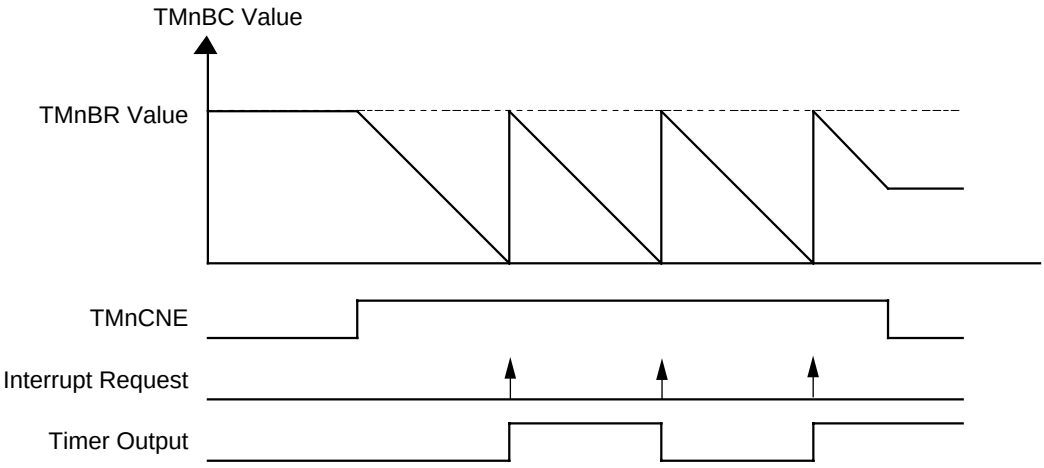


Figure 5-4-1 Interval Timer Operation

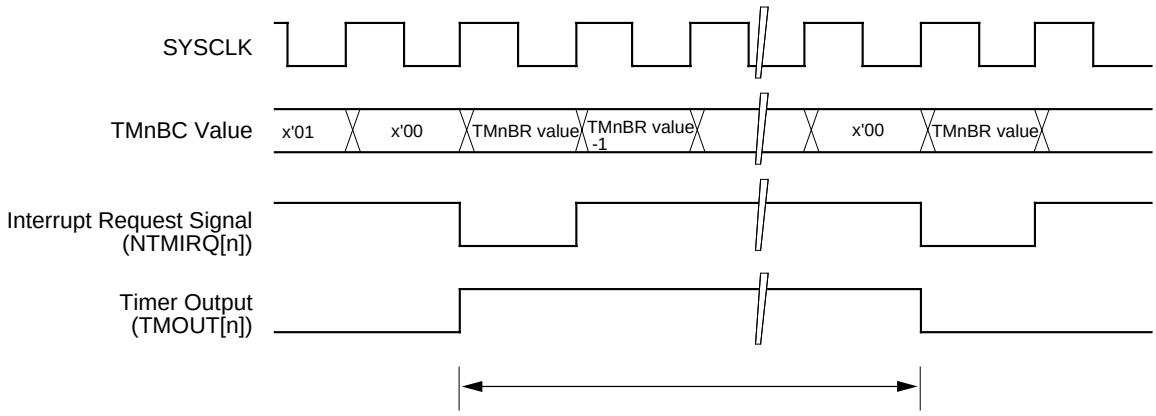


Figure 5-4-2 Interval Timer Operation (Clock Source = SYSCLK)

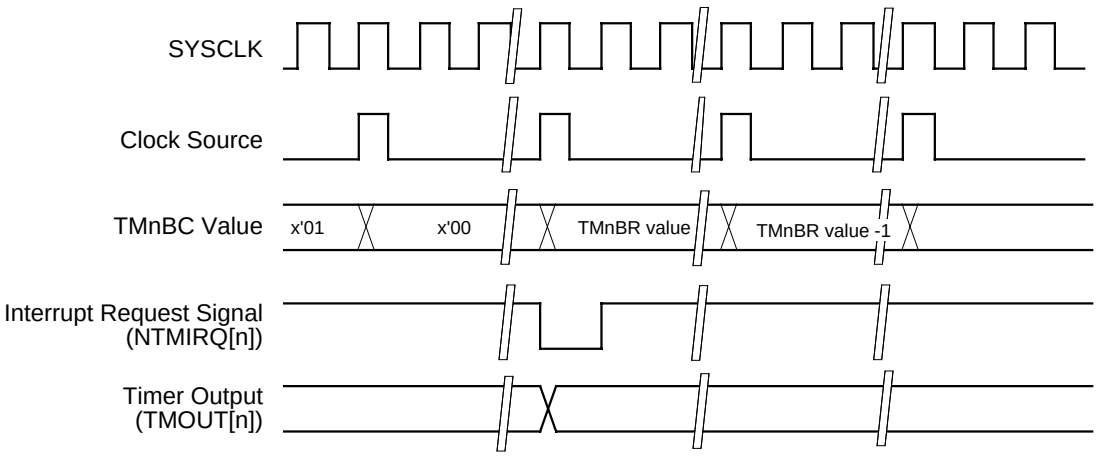


Figure 5-4-3 Interval Timer Operation (With Prescaler)

5-4-2 Interval Timer and Timer Output Setup Example

Table 5-4-1 List of Control Registers for Interval Timer and Timer Output

Register	R/W	Function
TMnBR	R/W	Timer n Base Register
TMnMD	R/W	Timer n Mode Register

n means timer number.

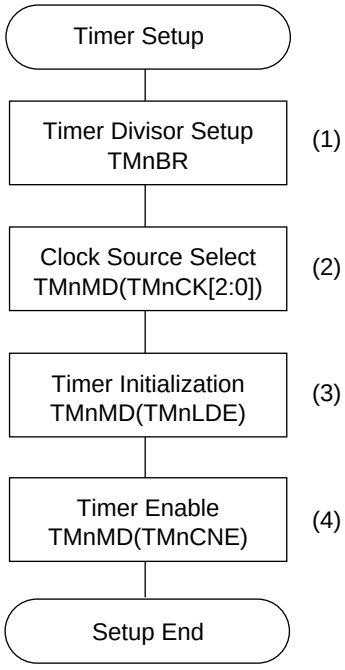
■ Timer Start

(1) Set the divisor of timer. Set the divisor to the TMnBR register. The valid range for TMnBR is 1 to 255. The interrupt request cycle should be (TMnBR set value +1) × Clock Source Cycle.

(2) Select Clock Source. Set the TMnCK[2:0] flags of the TMnMD register to the clock source. When selecting SYSCLK/8 or SYSCLK/32 as clock source, enable prescaler operation by setting the PS25CNE flag of the PSCNT25 register or PS69CNE flag of PSCNT69 register to 1.

(3) Initialize the timer by setting the TMnLDE of the TMnMD register to 1. Load the TMnBR value to the TMnBC counter as the initial value and reset the timer output. Set the TMnLDE to 0 and return to the normal operation always after initialization.

(4) Enable timer counting. Timer starts counting by setting the TMnCNE flag of the TMnMD register to 1.

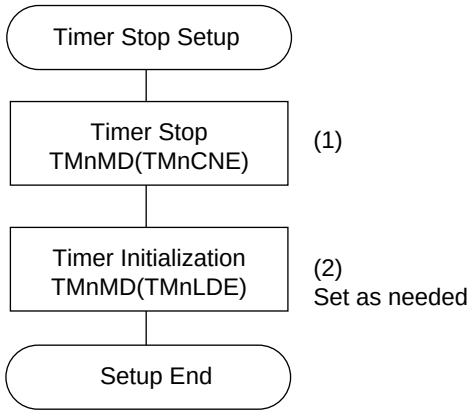


When enabling the counting, a timer underflow interrupt occurs at the fixed cycle. Whenever a timer underflow interrupt occurs, invert TMOUT[n] pin output and load the TMnBR value to the TMnBC counter. If changing the TMnBR value during counting, load the new TMnBR value to the TMnBC counter as the initial value when the next timer underflow interrupt occurs and change the interrupt cycle.

■ Timer Stop

(1) Stop the timer counting. Counting stops by setting the TMnCNE flag of the TMnMD register to 0.

(2) Initialize the timer as needed. When setting the TMnLDE of the TMnMD register to 1, load the TMnBR value to the TMnBC counter as the initial value and reset the timer output. When timer stops and TMnLDE is not set to 1, the TMnBC counter and pin output are remain as the status before timer stops. Setting TMnCNE to 1 starts counting from the status immediately before timer stops.



5-5 Event Count (Timer 2 to Timer 9)

5-5-1 Event Count

This section shows the setup procedures when using a 8-bit timer as an event counter.

> Please see “5-6 Cascade Connection” when cascading 8-bit timers forms a 16-bit timer, 24-bit timer or 32-bit timer.

■ Timer Start

(1) Set the divisor of timer. Set the divisor to the TMnBR register. The valid range for TMnBR is 1 to 255. An interrupt request occurs when the number of the rising edges of pin input reaches the TMnBR value plus 1.

(2) Select Clock Source. Set the TMnCK[2:0] flags of the TMnMD register to TMIN[n] pin input as the clock source.

(3) Initialize the timer by setting the TMnLDE of the TMnMD register to 1. Load the TMnBR value to the TMnBC counter as the initial value. Set the TMnLDE to 0 and return to the normal operation always after initialization.

> Set the TMnLDE flag while the TMnCNE flag is 0.

(4) Enable timer counting. Timer starts counting by setting the TMnCNE flag of the TMnMD register to 1.

When enabling the counting, the timer counts rising edges of TMIN[n] pin input and an interrupt request occurs at TMnBC underflow (See Figure 5-5-1). If changing the TMnBR value during counting, load the new TMnBR value to the TMnBC counter as the initial value when the next timer underflow interrupt occurs and change the interrupt cycle.

■ Timer Stop

(1) Stop the timer counting. Counting stops by setting the TMnCNE flag of the TMnMD register to 0.

(2) Initialize the timer as needed. When setting the TMnLDE of the TMnMD register to 1, load the TMnBR value to the TMnBC counter as the initial value. When timer stops and TMnLDE is not set to 1, the TMnBC counter remains the status before timer stops. Setting TMnCNE to 1 starts counting from the status immediately before timer stops.

Sample TMnIO pin input on SYSCLK. Input a signal with pulse width of SYSCLK × 1.5 or more. The event counter does not operate when SYSCLK stops (in STOP mode).

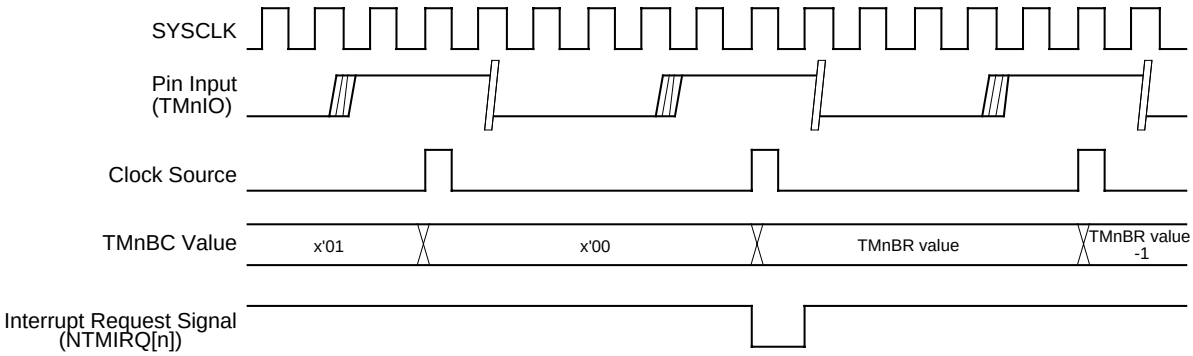


Figure 5-5-1 Event Count Operation

5-5-2 Event Counter Setup Example

Table 5-5-1 List of Control Registers for Event Counter

Register	R/W	Function
TMnBR	R/W	Timer n Base Register
TMnMD	R/W	Timer n Mode Register

n means timer number.

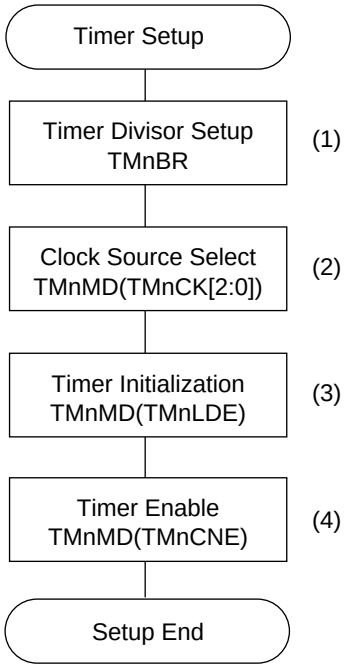
■ Timer Start

(1) Set the divisor of timer. Set the divisor to the TMnBR register. The valid range for TMnBR is 1 to 255. The interrupt request cycle should be (TMnBR set value +1) × Clock Source Cycle.

(2) Select Clock Source. Set the TMnCK[2:0] flags of the TMnMD register to the clock source. When selecting SYSCLK/8 or SYSCLK/32 as clock source, enable prescaler operation by setting the PS25CNE flag of the PSCNT25 register or PS69CNE flag of PSCNT69 register to 1.

(3) Initialize the timer by setting the TMnLDE of the TMnMD register to 1. Load the TMnBR value to the TMnBC counter as the initial value. Set the TMnLDE to 0 and return to the normal operation always after initialization.

(4) Enable timer counting. Timer starts counting by setting the TMnCNE flag of the TMnMD register to 1.

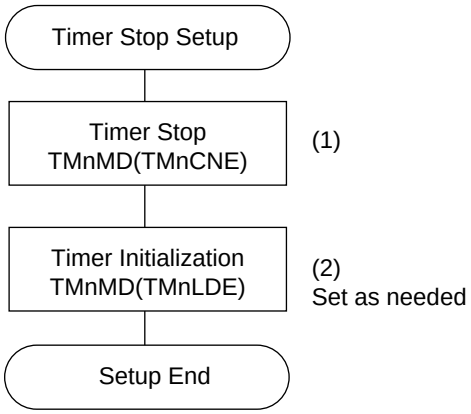


When enabling the counting, the timer counts rising edges of TMIN[n] pin input and an interrupt request occurs at TMnBC underflow. If changing the TMnBR value during counting, load the new TMnBR value to the TMnBC counter as the initial value when the next timer underflow interrupt occurs and change the interrupt cycle.

■ Timer Stop

(1) Stop the timer counting. Counting stops by setting the TMnCNE flag of the TMnMD register to 0.

(2) Initialize the timer as needed. When setting the TMnLDE of the TMnMD register to 1, load the TMnBR value to the TMnBC counter as the initial value. When timer stops and TMnLDE is not set to 1, the TMnBC counter remains the status before timer stops. Setting TMnCNE to 1 starts counting from the status immediately before timer stops.



Sample TMnIO pin input on SYSCLK. Input a signal with pulse width of SYSCLK × 1.5 or more. The event counter does not operate when SYSCLK stops (in STOP mode).

5-6 Cascade Connection (Timer 2 to Timer 9)

5-6-1 Cascade Operation

8-bit timers can be cascade as the following figure shows.

16-bit Timer

Upper	Lower
Timer 5	Timer 4
Timer 9	Timer 8

Upper	Lower
Timer 4	Timer 3
Timer 8	Timer 7

Upper	Lower
Timer 3	Timer 2
Timer 7	Timer 6

24-bit Timer

Upper		Lower
Timer 5	Timer 4	Timer 3
Timer 9	Timer 8	Timer 7

Upper		Lower
Timer 4	Timer 3	Timer 2
Timer 8	Timer 7	Timer 6

32-bit Timer

Upper			Lower
Timer 5	Timer 4	Timer 3	Timer 2
Timer 9	Timer 8	Timer 7	Timer 6

Figure 5-6-1 Cascade Connection

The following shows the 8-bit timer cascade procedure.

(1) Set the divisor of timer. Set the divisor to the TMnBR register.
In this example, cascade timer 2 and timer 3 to form a 16-bit timer and the interrupt interval is x'1234'. To do this, set the TMnBR to x'1234' - 1 (x'1233'. In this case, set the lower timer TM2BR to x'33' and the upper timer TM3BR to x'12'.

Since TMnBR can make a 16-bit access, values can be written to both registers at the same time. (When cascading timer 3 and timer 4 or timer 7 and timer 8, or timers are used as a 24-bit timer, values cannot be written to all registers at the same time.) The TMnBR value is changed during count operation only when timers are used as 8-bit timers or cascading timers except timer 5 and timer 6 are used as 16-bit timers. The TMnBR registers for timers to be used must be changed at the same time.

(2) Select Clock Source. Set the lower timer to its arbitrary clock source. Then select the upper timer to "cascade" as clock source.

Example 1: cascading timer 2 and timer 3 forms a 16-bit timer
Set the clock source to timer 2.
Set "cascade" to timer 3.

Example 2: cascading timer 2, timer 3 timer 4 and timer 5 forms a 32-bit timer
Set the clock source to timer 2.
Set "cascade" to timer 3, timer 4 and timer 5.

(3) Initialize timers by setting the TMnLDE flags of all cascaded timers to 1. (Setting all registers at the same time is not required.)

(4) Enable counting by using either of the following setup procedures.

1. Enable counting from the upper timer.
2. Enable counting of all timers at the same time.

(5) Disable counting by using either of the following setup procedures.

1. Disable counting from the lower timer.
2. Disable counting of all timers at the same time.

(6) Only the most upper timer of cascaded timers can generate timer output and interrupts.

Set interrupt disable although an interrupt request of the lower timer does not occur when cascading.

Difference when using timers as prescalers and when cascading timers

In this example, select the timer 2 underflow as the timer 3 clock source and select the timer 2 cascade as the timer 3 clock source. The figure 5-6-2 shows the operation when selecting the timer 2 underflow as the timer 3 clock source and selecting SYSCLK as the timer 2 clock source. When TM2BC underflows, the TM2BR value is loaded to TM2BC and the TM2BC value is subtracted by 1. When TM3BC underflows, the TM3BR value is loaded to TM3BC.

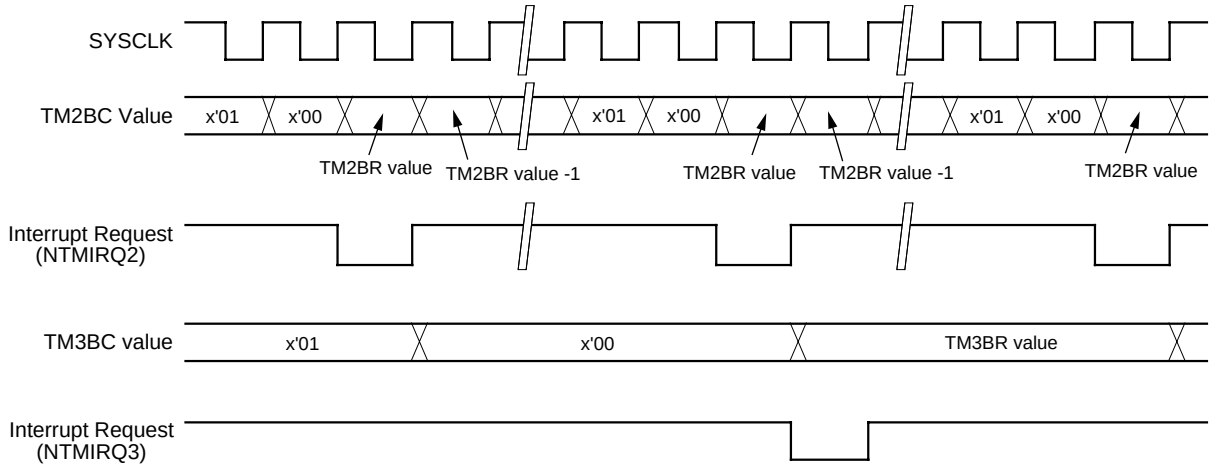


Figure 5-6-2 Timer 2, Timer 3 Operation 1

In this example, the figure 5-6-3 shows the operation when selecting the timer 2 cascade as the timer 3 clock source and selecting SYSCLK as the timer 2 clock source. When TM3BC is not x'00' and TM2BC underflows, the TM2BR value becomes x'FF' and the TM3BC value is subtracted by 1. When TM3BC is x'00' and TM2BC underflows, the TM2BR value and the TM3BR value is loaded to TM2BC and TM3BC respectively and then a timer 3 interrupt occurs.

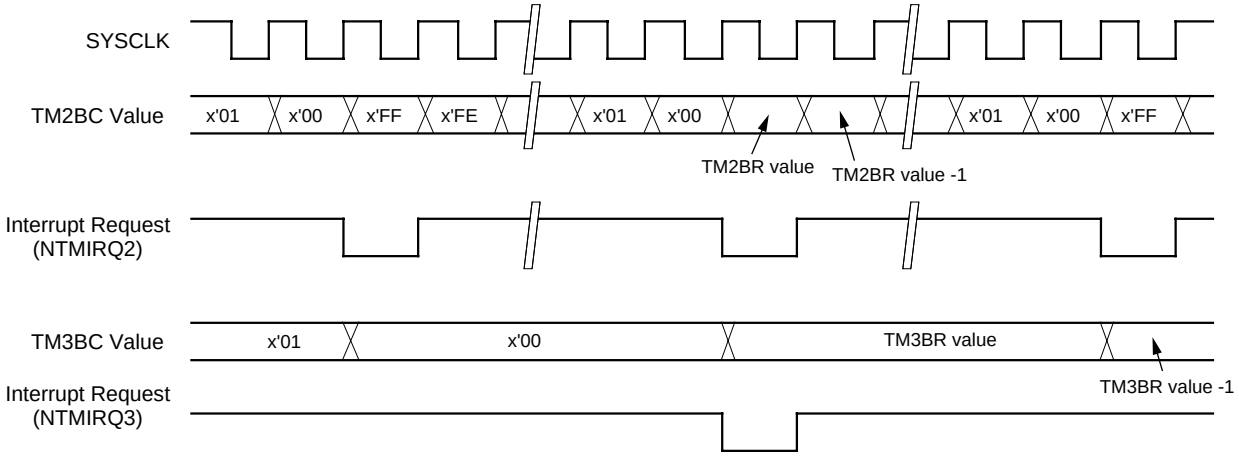


Figure 5-6-3 Timer 2, Timer 3 Operation 2

5-6-2 Cascade Setup Example

8-bit timers can cascade combining as figure 5-6-1 below.

Table 5-6-1 List of Control Registers for Cascade Connection

Register	R/W	Function
TMnBR	R/W	Timer n Base Register
TMnMD	R/W	Timer n Mode Register

n means timer number.

The following shows the 8-bit timer cascade procedure.

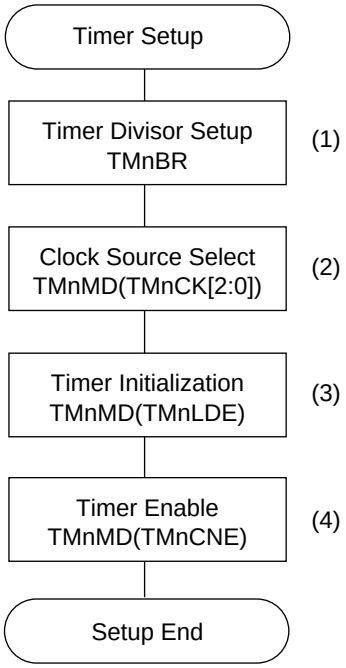
(1) Set the divisor of timer. Set the divisor to the TMnBR register.
In this example, cascade timer 2 and timer 3 to form a 16-bit timer and the interrupt interval is x'1234'. To do this, set the TMnBR to x'1234' - 1 (x'1233'. In this case, set the lower timer TM2BR to x'33' and the upper timer TM3BR to x'12'.

Since TMnBR can make a 16-bit access, values can be written to both registers at the same time. (When cascading timer 3 and timer 4 or timer 7 and timer 8, or timers are used as a 24-bit timer, values cannot be written to all registers at the same time.) The TMnBR value is changed during count operation only when timers are used as 8-bit timers or cascading timers except timer 5 and timer 6 are used as 16-bit timers. The TMnBR registers for timers to be used must be changed at the same time.

(2) Select Clock Source. Set the most lower timer to its arbitrary clock source. Then select the upper timer or timers to “cascade” as clock source.

Example 1: cascading timer 2 and timer 3 forms a 16-bit timer
Set the clock source to timer 2.
Set “cascade” to timer 3.

Example 2: cascading timer 2, timer 3 timer 4 and timer 5 forms a 32-bit timer
Set the clock source to timer 2.
Set “cascade” to timer 3, timer 4 and timer 5.



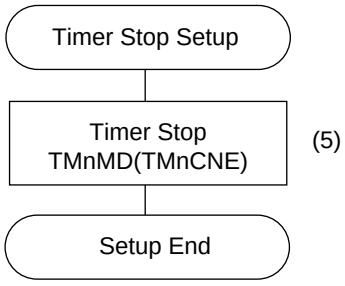
(3) Initialize timers by setting the TMnLDE flags of all cascaded timers to 1. (Setting all registers at the same time is not required.)

(4) Enable counting by using either of the following setup procedures.

1. Enable counting from the upper timer.
2. Enable counting of all timers at the same time.

(5) Disable counting by using either of the following setup procedures.

1. Disable counting from the lower timer.
2. Disable counting of all timers at the same time.



(6) Only the most upper timer of cascaded timers can generate timer output and interrupts.

5-7 Summary of 16-bit Timer

The MN102H74G/74F/74D/F74G contains four 16-bit timers. Each timer is an up/down counter and has two compare/capture registers.

5-7-1 List of 16-bit Timer Function

The following table shows functions of 16-bit timers.

Table 5-7-1 List of 16-bit Timer Functions

Function \ Timer	16-bit Timer			
	Timer 10	Timer 11	Timer 12	Timer 13
Interrupt Request	Subgroup 7 (SG70ICR) TM10IR TM10AIR TM10BIR	Subgroup 7(SG71ICR) TM11IR TM11AIR TM11BIR	Subgroup 8(SG80ICR) TM12IR TM12AIR TM12BIR	Subgroup 9(SG90ICR) TM13IR TM13AIR TM13BIR
Interrupt Source	Timer 10 Over/Underflow Timer 10 Compare/Capture A Timer 10 Compare/Capture B	Timer 11 Over/Underflow Timer 11 Compare/Capture A Timer 11 Compare/Capture B	Timer 12 Over/Underflow Timer 12 Compare/Capture A Timer 12 Compare/Capture B	Timer 13 Over/Underflow Timer 13 Compare/Capture A Timer 13 Compare/Capture B
Clock Source	SYSCLK SYSCLK/8 Timer 2 Underflow Timer 3 Underflow Two-phase Encoder (×1) Two-phase Encoder (×4) TM10IOB pin (Both Edges) TM10IOB pin (Positive Edge/Negative Edge)	SYSCLK SYSCLK/8 Timer 8 Underflow Timer 9 Underflow Two-phase Encoder (×1) Two-phase Encoder (×4) TM11IOB pin (Both Edges) TM11IOB pin (Positive Edge/Negative Edge)	SYSCLK SYSCLK/8 Timer 4 Underflow Timer 5 Underflow Two-phase Encoder (×1) Two-phase Encoder (×4) TM12IOB pin (Both Edges) TM12IOB pin (Positive Edge/Negative Edge)	SYSCLK SYSCLK/8 Timer 6 Underflow Timer 7 Underflow Two-phase Encoder (×1) Two-phase Encoder (×4) TM13IOB pin (Both Edges) TM13IOB pin (Positive Edge/Negative Edge)
Counter	Up/Down Counter	Up/Down Counter	Up/Down Counter	Up/Down Counter
Interval Timer	✓	✓	✓	✓
Event Counter	✓ Edge Selectable	✓ Edge Selectable	✓ Edge Selectable	✓ Edge Selectable
PWM Output	✓ Cycle, Duty Changeable	✓ Cycle, Duty Changeable	✓ Cycle, Duty Changeable	✓ Cycle, Duty Changeable
Input Capture	✓ Positive, Negative or Both Edges Selectable	✓ Positive, Negative or Both Edges Selectable	✓ Positive, Negative or Both Edges Selectable	✓ Positive, Negative or Both Edges Selectable
One-shot Output	✓	✓	✓	✓
External Trigger Activation	✓	✓	✓	✓
Timer Output	✓	✓	✓	✓

Note 1: Set the prescaler control register PSCNTn (n=10 to 13) to enable when selecting SYSCLK/8.
Note 2: Count the edge selected by the TMnBEG flags of TMnMDB register when selecting the positive edge or negative edge of the TMnIOB pin.

5-7-2 16-bit Timer Block Diagram

This section shows the block diagram of 16-bit timer.

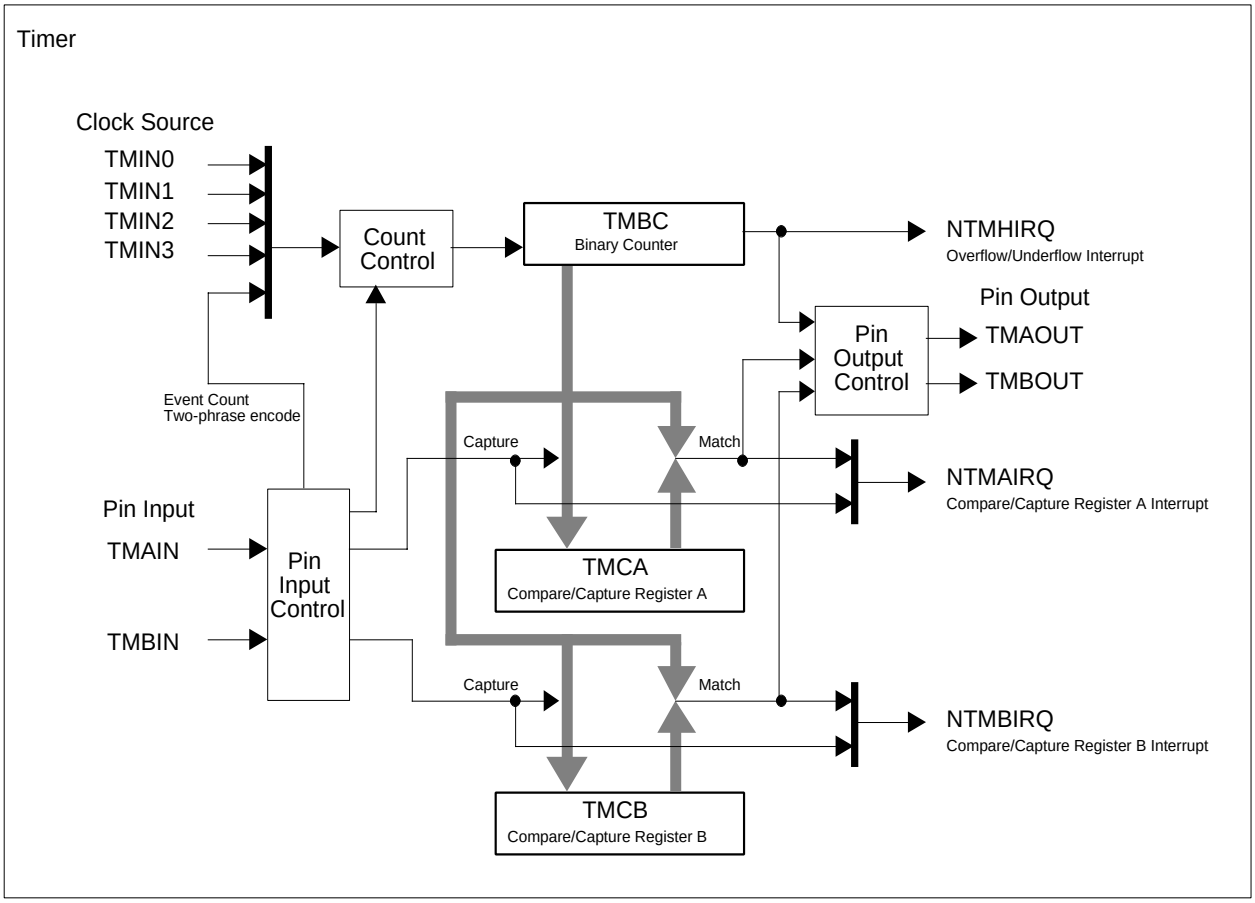


Figure 5-7-1 16-bit Timer Block Diagram

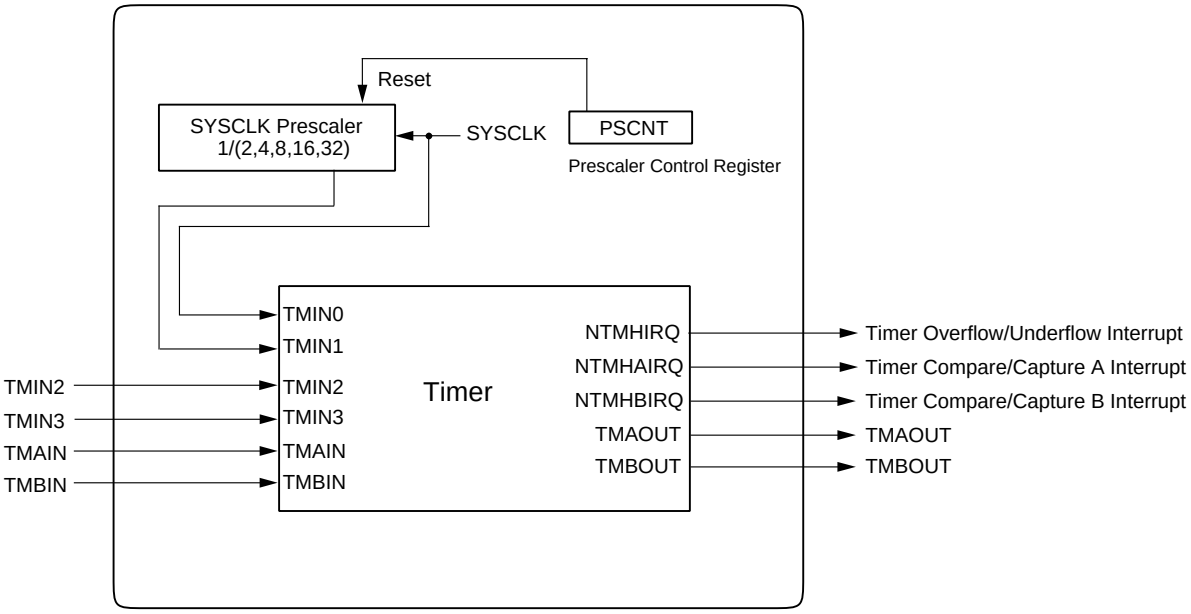


Figure 5-7-2 16-bit Timer Connection

5-8 16-bit Timer Control Register

5-8-1 List of Timer 10 to Timer 13 Control Registers

The following table lists 16-bit timer control registers.

Table 5-8-1 List of 16-bit Timer Control Registers

Register	Address	R/W	Function
PSCNT10	x'E00100'	R/W	Prescaler 10 Control Register
TM10MD	x'E00110'	R/W	Timer 10 Mode Register
TM10MDA	x'E00112'	R/W	Timer 10 Compare Capture A Mode Register
TM10MDB	x'E00113'	R/W	Timer 10 Compare Capture B Mode Register
TM10CA	x'E00114'	R/W	Timer 10 Compare Capture A Register
TM10CB	x'E00116'	R/W	Timer 10 Compare Capture B Register
TM10BC	x'E00118'	R	Timer 10 Binary Counter
PSCNT11	x'E00120'	R/W	Prescaler 11 Control Register
TM11MD	x'E00130'	R/W	Timer 11 Mode Register
TM11MDA	x'E00132'	R/W	Timer 11 Compare Capture A Mode Register
TM11MDB	x'E00133'	R/W	Timer 11 Compare Capture B Mode Register
TM11CA	x'E00134'	R/W	Timer 11 Compare Capture A Register
TM11CB	x'E00136'	R/W	Timer 11 Compare Capture B Register
TM11BC	x'E00138'	R	Timer 11 Binary Counter
PSCNT12	x'E00140'	R/W	Prescaler 12 Control Register
TM12MD	x'E00150'	R/W	Timer 12 Mode Register
TM12MDA	x'E00152'	R/W	Timer 12 Compare Capture A Mode Register
TM12MDB	x'E00153'	R/W	Timer 12 Compare Capture B Mode Register
TM12CA	x'E00154'	R/W	Timer 12 Compare Capture A Register
TM12CB	x'E00156'	R/W	Timer 12 Compare Capture B Register
TM12BC	x'E00158'	R	Timer 12 Binary Counter
PSCNT13	x'E00160'	R/W	Prescaler 13 Control Register
TM13MD	x'E00170'	R/W	Timer 13 Mode Register
TM13MDA	x'E00172'	R/W	Timer 13 Compare Capture A Mode Register
TM13MDB	x'E00173'	R/W	Timer 13 Compare Capture B Mode Register
TM13CA	x'E00174'	R/W	Timer 13 Compare Capture A Register
TM13CB	x'E00176'	R/W	Timer 13 Compare Capture B Register
TM13BC	x'E00178'	R	Timer 13 Binary Counter

5-8-2 Timer 10 to Timer 13 Programmable Timer Registers

■ Timer Compare/Capture A Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TM CA15	TM CA14	TM CA13	TM CA12	TM CA11	TM CA10	TM CA9	TM CA8	TM CA7	TM CA6	TM CA5	TM CA4	TM CA3	TM CA2	TM CA1	TM CA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5-8-1 Timer Compare/Capture A Register (TMnCA, R/W)

Register	Address
TM10CA	x'E00114'
TM11CA	x'E00134'
TM12CA	x'E00154'
TM13CA	x'E00174'

■ Timer Compare Register

When TMnBC matches TMnCA, the interrupt request NTMAIRQ occurs. Timer divisor is set by setting TMnBC clear when TMnBC matches TMnCA. The divisor is TMnCA value +1. When the TMnCA is set to double buffer, the TMnCA value is saved in the TMnCA buffer once and TMnBC may read previous setting value after TMnCA is written. The set value is loaded from TMnCA buffer to TMnCA in the following conditions. In either condition, TMnBC is x'0000'.

1. Timer initialization
2. Timer overflow/Timer underflow (When TMnCLE is 0.)
3. Timer is up counting when TMnBC matches TMnCA (When TMnCLE is 1).

■ Timer Capture Register

When the edge selected by TMAEG is input to TMAIN pin, the TMnBC value is captured to TMnCA and the interrupt request NTMAIRQ occurs. When TMnCA is set to both edges, TMnCA captures regardless of the rising edge or the falling edge and an interrupt request occurs.

The Figure 5-8-3 shows the block diagram of the timer compare/capture registers.

■ Timer Compare/Capture B Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TM CB15	TM CB14	TM CB13	TM CB12	TM CB11	TM CB10	TM CB9	TM CB8	TM CB7	TM CB6	TM CB5	TM CB4	TM CB3	TM CB2	TM CB1	TM CB0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5-8-2 Timer Compare/Capture B Register (TMnCB, R/W)

Register	Address
TM10CB	x'E00116'
TM11CB	x'E00136'
TM12CB	x'E00156'
TM13CB	x'E00176'

■ Timer Compare Register

When TMnBC matches TMnCB, the interrupt request NTMBIRQ occurs. When the TMnCB is set to double buffer, the TMnCB value is saved in the TMnCB buffer once and TMnBC may read previous setting value after TMnCB is written. The set value is loaded from TMnCB buffer to TMnCB in the following conditions. In either condition, TMnBC is x'0000'.

1. Timer initialization
2. Timer overflow/Timer underflow (When TMnCLE is 0.)
3. Timer is up counting when TMnBC matches TMnCB (When TMnCA is selected as compare register and TMnCLE is 1).
4. TMnCA capture (When TMnCA is selected as capture register and TMnCLE is 1.)

■ Timer Capture Register

When the edge selected by TMBEG is input to TMBIN pin, the TMnBC value is captured to TMnCB and the interrupt request NTMBIRQ occurs. When TMnCB is set to both edges, TMnCB captures regardless of the rising edge or the falling edge and an interrupt request occurs.

The Figure 5-8-3 shows the block diagram of the timer compare/capture registers.

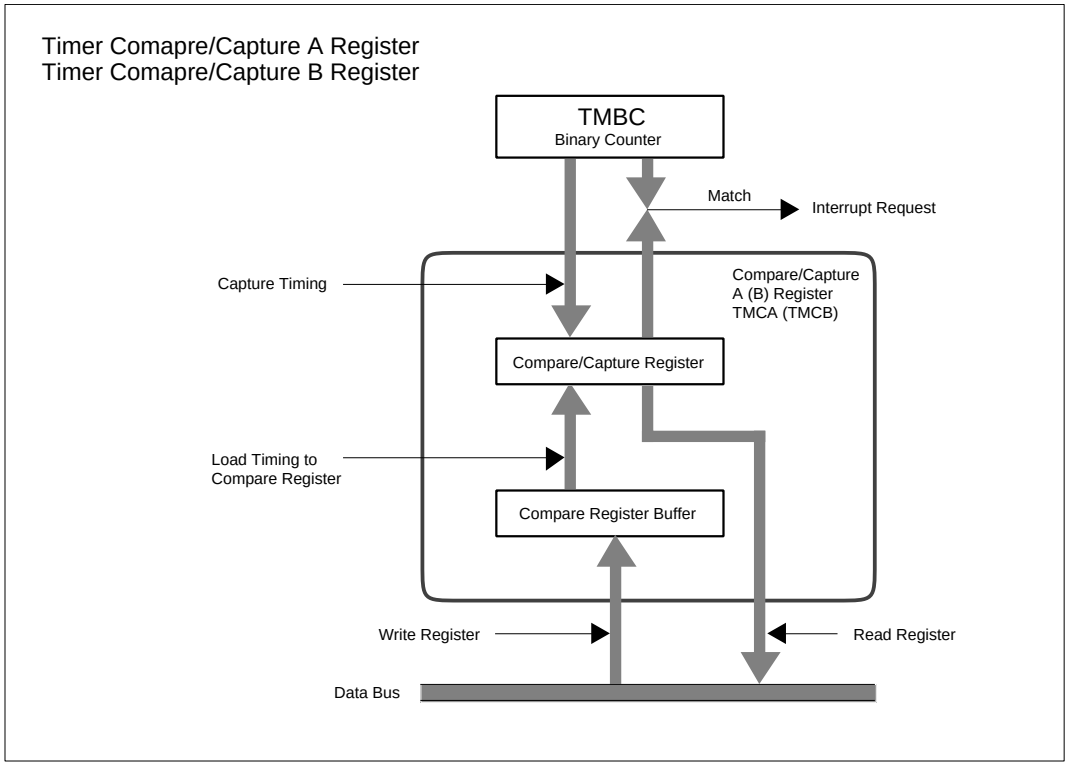


Figure 5-8-3 Timer Compare/Capture Register Block Diagram

■ Timer Binary Counter

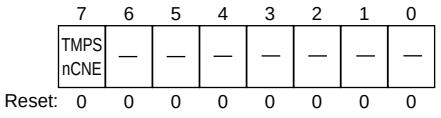
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TM BC15	TM BC14	TM BC13	TM BC12	TM BC11	TM BC10	TM BC9	TM BC8	TM BC7	TM BC6	TM BC5	TM BC4	TM BC3	TM BC2	TM BC1	TM BC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5-8-4 Timer Binary Counter (TMnBC, R/W)

Register	Address
TM10BC	x'E00118'
TM11BC	x'E00138'
TM12BC	x'E00158'
TM13BC	x'E00178'

TMnBC reads the counting value. TMnBC is an up/down counter. TMnBC up counting or down counting from the initial value x'0000'. An interrupt request occurs when TMnBC overflows or underflows.

■ Prescaler Control Register



TMPSnCNE	Prescaler Count Enable
0	Disable
1	Enable

Figure 5-8-5 Prescaler Control Register (PSCNTn, R/W)

Register	Address
PSCNT10	x'E00100'
PSCNT11	x'E00120'
PSCNT12	x'E00140'
PSCNT13	x'E00160'

5-8-3 Timer 10 to Timer 13 Mode Registers

■ Timer 10 Mode Register

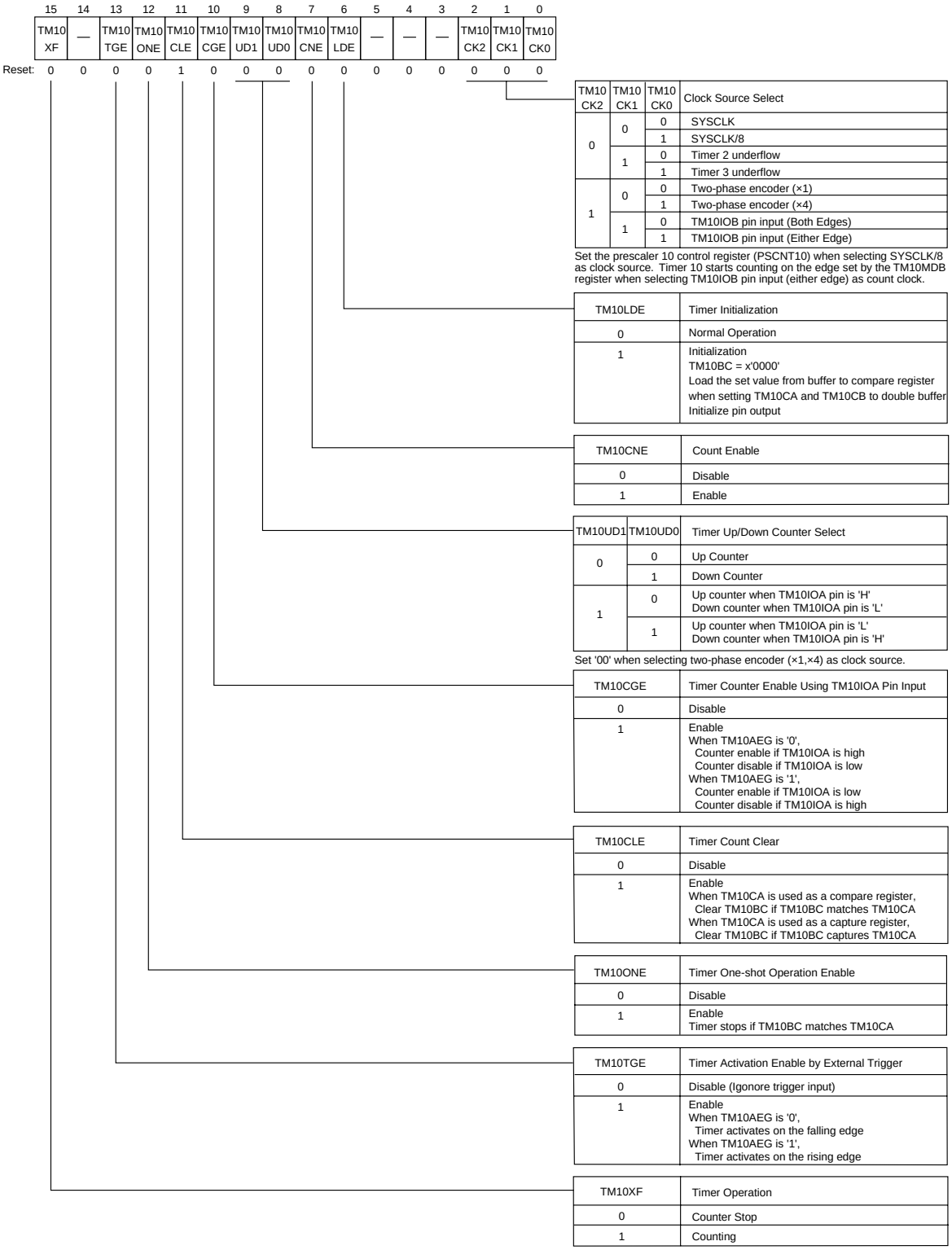

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TM10XF — TM10TGE TM10ONE TM10CLE TM10CGE TM10UD1 TM10UD0 TM10CNE TM10LDE — — — TM10CK2 TM10CK1 TM10CK0
Reset: 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
TM10CK2 TM10CK1 TM10CK0 Clock Source Select
0 0 0 SYSCLK
0 SYSCLK/8
1 0 Timer 2 underflow
1 Timer 3 underflow
1 0 0 Two-phase encoder (×1)
1 Two-phase encoder (×4)
1 0 TM10IOB pin input (Both Edges)
1 TM10IOB pin input (Either Edge)
Set the prescaler 10 control register (PSCNT10) when selecting SYSCLK/8 as clock source. Timer 10 starts counting on the edge set by the TM10MDB register when selecting TM10IOB pin input (either edge) as count clock.
TM10LDE Timer Initialization
0 Normal Operation
1 Initialization
TM10BC = x'0000'
Load the set value from buffer to compare register when setting TM10CA and TM10CB to double buffer
Initialize pin output
TM10CNE Count Enable
0 Disable
1 Enable
TM10UD1 TM10UD0 Timer Up/Down Counter Select
0 0 Up Counter
1 Down Counter
1 0 Up counter when TM10IOA pin is 'H'
Down counter when TM10IOA pin is 'L'
1 Up counter when TM10IOA pin is 'L'
Down counter when TM10IOA pin is 'H'
Set '00' when selecting two-phase encoder (×1,×4) as clock source.
TM10CGE Timer Counter Enable Using TM10IOA Pin Input
0 Disable
1 Enable
When TM10AEG is '0',
Counter enable if TM10IOA is high
Counter disable if TM10IOA is low
When TM10AEG is '1',
Counter enable if TM10IOA is low
Counter disable if TM10IOA is high
TM10CLE Timer Count Clear
0 Disable
1 Enable
When TM10CA is used as a compare register,
Clear TM10BC if TM10BC matches TM10CA
When TM10CA is used as a capture register,
Clear TM10BC if TM10BC captures TM10CA
TM10ONE Timer One-shot Operation Enable
0 Disable
1 Enable
Timer stops if TM10BC matches TM10CA
TM10TGE Timer Activation Enable by External Trigger
0 Disable (Igonore trigger input)
1 Enable
When TM10AEG is '0',
Timer activates on the falling edge
When TM10AEG is '1',
Timer activates on the rising edge
TM10XF Timer Operation
0 Counter Stop
1 Counting

Figure 5-8-6 Timer 10 Mode Register (TM10MD: x'E00110', R/W)

■ Timer 11 Mode Register

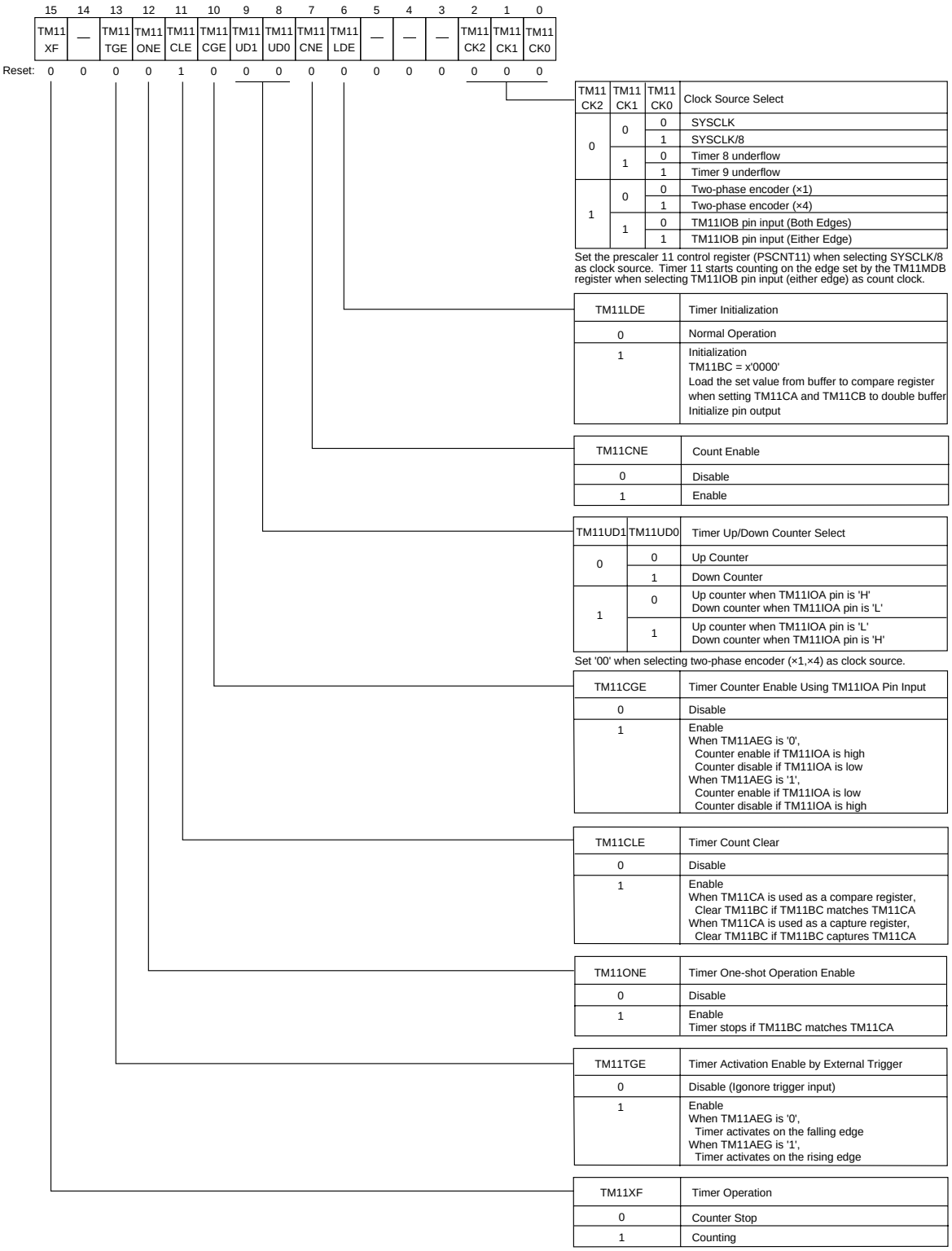

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TM11XF — TM11TGE TM11ONE TM11CLE TM11CGE TM11UD1 TM11UD0 TM11CNE TM11LDE — — — TM11CK2 TM11CK1 TM11CK0
Reset: 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
TM11CK2 TM11CK1 TM11CK0 Clock Source Select
0 0 0 SYSCLK
1 SYSCLK/8
1 0 Timer 8 underflow
1 Timer 9 underflow
1 0 0 Two-phase encoder (×1)
1 Two-phase encoder (×4)
1 0 TM11IOB pin input (Both Edges)
1 TM11IOB pin input (Either Edge)
Set the prescaler 11 control register (PSCNT11) when selecting SYSCLK/8 as clock source. Timer 11 starts counting on the edge set by the TM11MDB register when selecting TM11IOB pin input (either edge) as count clock.
TM11LDE Timer Initialization
0 Normal Operation
1 Initialization
TM11BC = x'0000'
Load the set value from buffer to compare register when setting TM11CA and TM11CB to double buffer
Initialize pin output
TM11CNE Count Enable
0 Disable
1 Enable
TM11UD1 TM11UD0 Timer Up/Down Counter Select
0 0 Up Counter
1 Down Counter
1 0 Up counter when TM11IOA pin is 'H'
Down counter when TM11IOA pin is 'L'
1 Up counter when TM11IOA pin is 'L'
Down counter when TM11IOA pin is 'H'
Set '00' when selecting two-phase encoder (×1,×4) as clock source.
TM11CGE Timer Counter Enable Using TM11IOA Pin Input
0 Disable
1 Enable
When TM11AEG is '0',
Counter enable if TM11IOA is high
Counter disable if TM11IOA is low
When TM11AEG is '1',
Counter enable if TM11IOA is low
Counter disable if TM11IOA is high
TM11CLE Timer Count Clear
0 Disable
1 Enable
When TM11CA is used as a compare register,
Clear TM11BC if TM11BC matches TM11CA
When TM11CA is used as a capture register,
Clear TM11BC if TM11BC captures TM11CA
TM11ONE Timer One-shot Operation Enable
0 Disable
1 Enable
Timer stops if TM11BC matches TM11CA
TM11TGE Timer Activation Enable by External Trigger
0 Disable (Igonore trigger input)
1 Enable
When TM11AEG is '0',
Timer activates on the falling edge
When TM11AEG is '1',
Timer activates on the rising edge
TM11XF Timer Operation
0 Counter Stop
1 Counting

Figure 5-8-7 Timer 11 Mode Register (TM11MD: x'E00130', R/W)

■ Timer 12 Mode Register

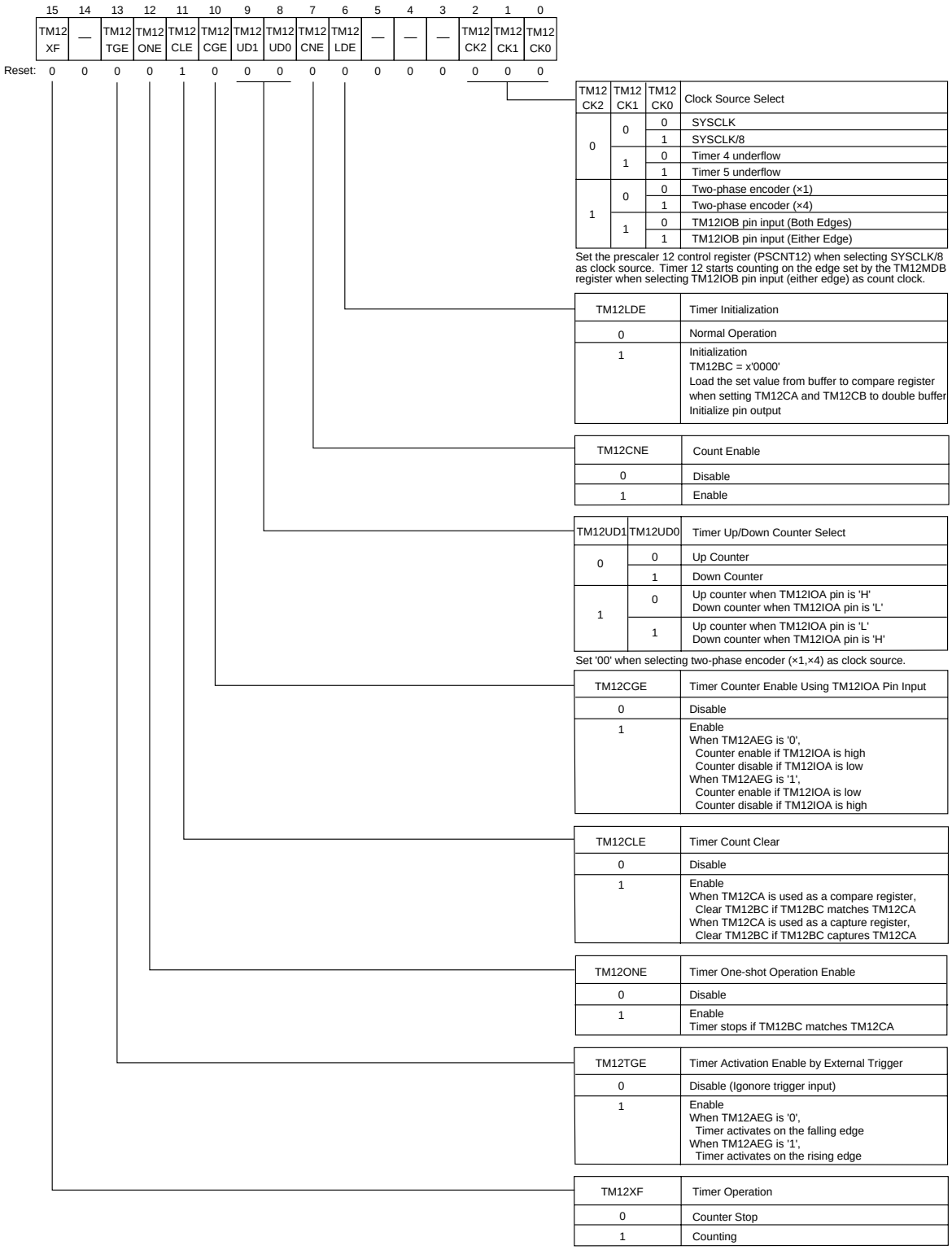

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TM12XF — TM12TGE TM12ONE TM12CLE TM12CGE TM12UD1 TM12UD0 TM12CNE TM12LDE — — — TM12CK2 TM12CK1 TM12CK0
Reset: 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
TM12CK2 TM12CK1 TM12CK0 Clock Source Select
0 0 0 SYSCLK
1 SYSCLK/8
1 0 Timer 4 underflow
1 Timer 5 underflow
1 0 0 Two-phase encoder (×1)
1 Two-phase encoder (×4)
1 0 TM12IOB pin input (Both Edges)
1 TM12IOB pin input (Either Edge)
Set the prescaler 12 control register (PSCNT12) when selecting SYSCLK/8 as clock source. Timer 12 starts counting on the edge set by the TM12MDB register when selecting TM12IOB pin input (either edge) as count clock.
TM12LDE Timer Initialization
0 Normal Operation
1 Initialization
TM12BC = x'0000'
Load the set value from buffer to compare register when setting TM12CA and TM12CB to double buffer
Initialize pin output
TM12CNE Count Enable
0 Disable
1 Enable
TM12UD1 TM12UD0 Timer Up/Down Counter Select
0 0 Up Counter
1 Down Counter
1 0 Up counter when TM12IOA pin is 'H'
Down counter when TM12IOA pin is 'L'
1 Up counter when TM12IOA pin is 'L'
Down counter when TM12IOA pin is 'H'
Set '00' when selecting two-phase encoder (×1,×4) as clock source.
TM12CGE Timer Counter Enable Using TM12IOA Pin Input
0 Disable
1 Enable
When TM12AEG is '0',
Counter enable if TM12IOA is high
Counter disable if TM12IOA is low
When TM12AEG is '1',
Counter enable if TM12IOA is low
Counter disable if TM12IOA is high
TM12CLE Timer Count Clear
0 Disable
1 Enable
When TM12CA is used as a compare register,
Clear TM12BC if TM12BC matches TM12CA
When TM12CA is used as a capture register,
Clear TM12BC if TM12BC captures TM12CA
TM12ONE Timer One-shot Operation Enable
0 Disable
1 Enable
Timer stops if TM12BC matches TM12CA
TM12TGE Timer Activation Enable by External Trigger
0 Disable (Igonore trigger input)
1 Enable
When TM12AEG is '0',
Timer activates on the falling edge
When TM12AEG is '1',
Timer activates on the rising edge
TM12XF Timer Operation
0 Counter Stop
1 Counting

Figure 5-8-8 Timer 12 Mode Register (TM12MD: x'E00150', R/W)

■ Timer 13 Mode Register

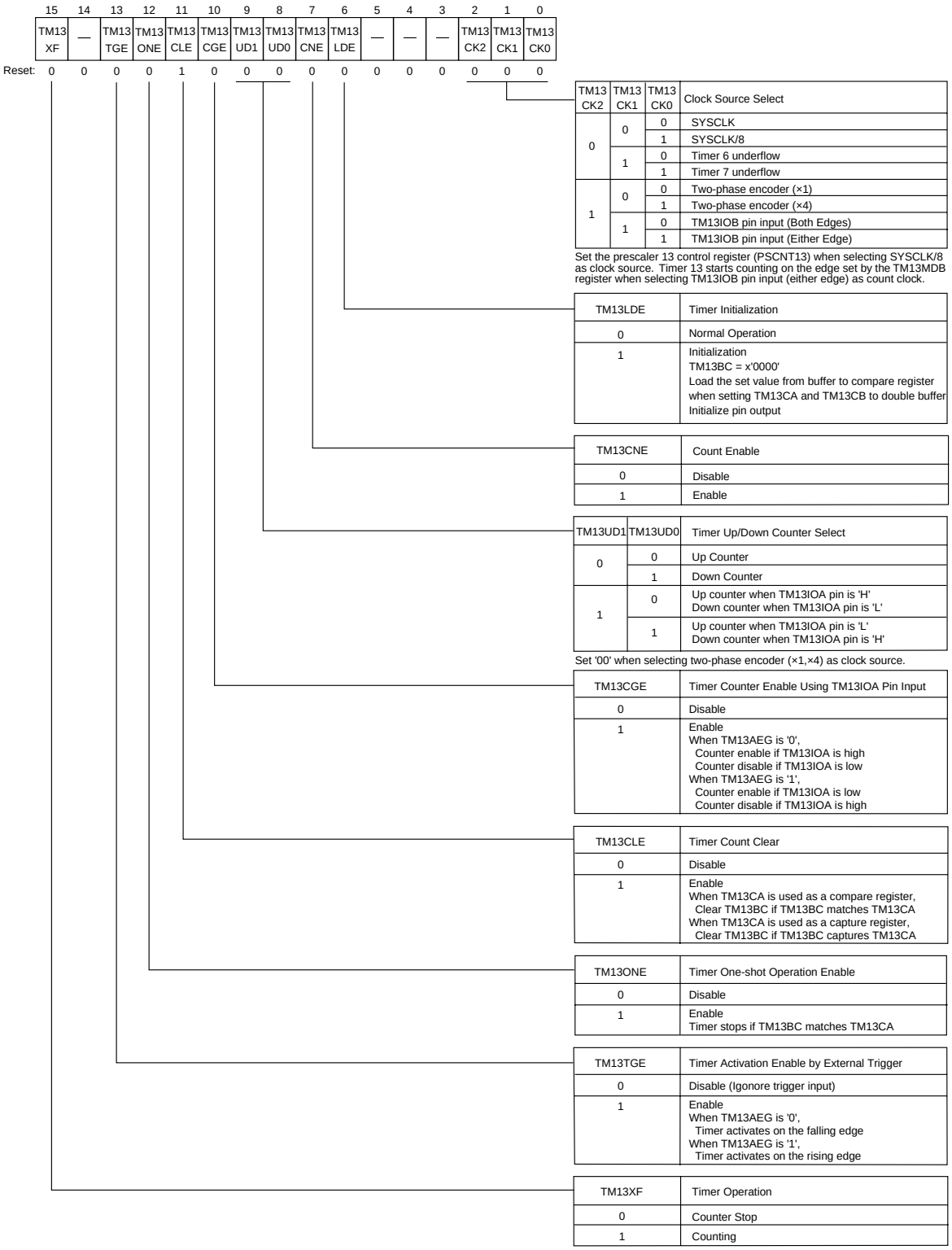

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TM13XF — TM13TGE TM13ONE TM13CLE TM13CGE TM13UD1 TM13UD0 TM13CNE TM13LDE — — — TM13CK2 TM13CK1 TM13CK0
Reset: 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
TM13CK2 TM13CK1 TM13CK0 Clock Source Select
0 0 0 SYSCLK
0 SYSCLK/8
1 0 Timer 6 underflow
1 Timer 7 underflow
1 0 0 Two-phase encoder (×1)
1 Two-phase encoder (×4)
1 0 TM13IOB pin input (Both Edges)
1 TM13IOB pin input (Either Edge)
Set the prescaler 13 control register (PSCNT13) when selecting SYSCLK/8 as clock source. Timer 13 starts counting on the edge set by the TM13MDB register when selecting TM13IOB pin input (either edge) as count clock.
TM13LDE Timer Initialization
0 Normal Operation
1 Initialization
TM13BC = x'0000'
Load the set value from buffer to compare register when setting TM13CA and TM13CB to double buffer
Initialize pin output
TM13CNE Count Enable
0 Disable
1 Enable
TM13UD1 TM13UD0 Timer Up/Down Counter Select
0 0 Up Counter
1 Down Counter
1 0 Up counter when TM13IOA pin is 'H'
Down counter when TM13IOA pin is 'L'
1 Up counter when TM13IOA pin is 'L'
Down counter when TM13IOA pin is 'H'
Set '00' when selecting two-phase encoder (×1,×4) as clock source.
TM13CGE Timer Counter Enable Using TM13IOA Pin Input
0 Disable
1 Enable
When TM13AEG is '0',
Counter enable if TM13IOA is high
Counter disable if TM13IOA is low
When TM13AEG is '1',
Counter enable if TM13IOA is low
Counter disable if TM13IOA is high
TM13CLE Timer Count Clear
0 Disable
1 Enable
When TM13CA is used as a compare register,
Clear TM13BC if TM13BC matches TM13CA
When TM13CA is used as a capture register,
Clear TM13BC if TM13BC captures TM13CA
TM13ONE Timer One-shot Operation Enable
0 Disable
1 Enable
Timer stops if TM13BC matches TM13CA
TM13TGE Timer Activation Enable by External Trigger
0 Disable (Igonore trigger input)
1 Enable
When TM13AEG is '0',
Timer activates on the falling edge
When TM13AEG is '1',
Timer activates on the rising edge
TM13XF Timer Operation
0 Counter Stop
1 Counting

Figure 5-8-9 Timer 13 Mode Register (TM13MD: x'E00170', R/W)

■ Timer Compare/Capture A Mode Register

The timer compare/capture A mode register controls the timer compare/capture A register and sets the waveform output to the TMnIOA pin.

7	6	5	4	3	2	1	0
TMn AM1	TMn AM0	TMn AEG	TMn ACE	—	—	TMn AO1	TMn AO0
Reset: 0	0	0	0	0	0	0	0

TMn AO1	TMn AO0	Timer A Output Waveform Select
0	0	High when TMnBC matches TMnCA Low when TMnBC matches TMnCB
	1	Set when TMnBC matches TMnCA Reset when TMnBC overflows
1	0	Set when TMnBC matches TMnCA (Reset only when timer is initialized.)
	1	Toggle output (Invert when TMnBC matches TMnCA)

TMnACE	Timer Capture A Operation Enable
0	Disable (Ignore pin input)
1	Enable

TMnAEG	Timer A Pin Polarity Select
0	Capture : Rising Edge (Either Edge Select) Count Control : Counting when TMnIOA is high Activation Trigger : Falling Edge Pin Output : Positive Output (Normal Output) (TMnIOA pin is low when the pin output controller is reset, TMnIOA pin is high when the pin output controller is set.)
1	Capture : Falling Edge (Either Edge Select) Count Control : Counting when TMnIOA is low Activation Trigger : Rising Edge Pin Output : Negative Output (Invert Output) (TMnIOA pin is high when the pin output controller is reset, TMnIOA pin is low when the pin output controller is set.)

TMn AM1	TMn AM0	Timer Compare/Capture A Operating Mode Select
0	0	Compare Register (Double Buffer)
	1	Compare Register (Single Buffer)
1	0	Capture Register (Positive/Negative Edge Select) Capture on the edge selected by TMnAEG
	1	Capture Register (Positive/Negative Edge Select) Ignore TMnAEG select

Figure 5-8-10 Timer Compare/Capture A Mode Register (TMnMDA, R/W)

Register	Address
TM10MDA	x'E00112'
TM11MDA	x'E00132'
TM12MDA	x'E00152'
TM13MDA	x'E00172'

■ Timer Compare/Capture B Mode Register

The timer compare/capture B mode register controls the timer compare/capture B register and sets the waveform output to the TMnIOB pin.

7	6	5	4	3	2	1	0
TMn BM1	TMn BM0	TMn BEG	TMn BCE	—	—	TMn BO1	TMn BO0
Reset: 0	0	0	0	0	0	0	0

TMn BO1	TMn BO0	Timer B Output Waveform Select
0	0	High when TMnBC matches TMnCB Low when TMnBC matches TMnCA
	1	Set when TMnBC matches TMnCB Reset when TMnBC overflows
1	0	Set when TMnBC matches TMnCB (Reset only when timer is initialized.)
	1	Toggle output (Invert when TMnBC matches TMnCB)

TMnBCE	Timer Capture B Operation Enable
0	Disable (Ignore pin input)
1	Enable

TMnBEG	TMnIOB Pin Polarity Select
0	Capture : Rising Edge (Either Edge Select) Count Source Input : Rising Edge (Either Edge Select) Pin Output : Positive Output (Normal Output) (TMnIOB pin is low when the pin output controller is reset, TMnIOB pin is high when the pin output controller is set.)
1	Capture : Falling Edge (Either Edge Select) Count Source Input : Rising Edge (Either Edge Select) Pin Output : Negative Output (Invert Output) (TMnIOB pin is high when the pin output controller is reset, TMnIOB pin is low when the pin output controller is set.)

TMn BM1	TMn BM0	Timer Compare/Capture B Operating Mode Select
0	0	Compare Register (Double Buffer)
	1	Compare Register (Single Buffer)
1	0	Capture Register (Positive/Negative Edge Select) Capture on the edge selected by TMnBEG
	1	Capture Register (Positive/Negative Edge Select) Ignore TMnBEG select

Figure 5-8-11 Timer Compare/Capture B Mode Register (TMnMDB, R/W)

Register	Address
TM10MDB	x'E00113'
TM11MDB	x'E00133'
TM12MDB	x'E00153'
TM13MDB	x'E00173'

5-9 16-bit Timer Operation

5-9-1 16-bit Timer Operation Examples

This section explains 16-bit timer operations. Each timer is an up/down counter and has two compare/capture registers. Each compare/capture register is used as a compare register or a capture register independently.

■ Binary Counter Setup

The timer binary counter (TMnBC) is controlled by the timer mode register (TMnMD). This example explains the timer mode register.

(1) Clock Source Select

Select the clock source by setting the TMnCK[2:0] flags. When selecting SYSCLK/8 as clock source, set the prescaler by the prescaler control register (PSCNTn) before timer starts. When selecting a two-phase encoder, timer counts up or down on the conditions shown in Table 5-9-1 and Table 5-9-2.

Table 5-9-1 Counting Operation When Two-phase Encoder (×1) is Selected

Up/Down Counter	Up Counter	Down Counter
TMnIOA Pin Input	"H"	"H"
TMnIOB Pin Input		

Table 5-9-2 Counting Operation When Two-phase Encoder (×4) is Selected

Up/Down Counter	Up Counter				Down Counter			
TMnIOA Pin Input		"L"		"H"		"L"		"H"
TMnIOB Pin Input	"H"		"L"		"L"		"H"	

(2) Timer Initialization

Initialize the timer by setting TMnLDE flag to ‘1’ immediately before enabling the timer after setting each register. At this point, do not change other flags. The TMnLDE flag must be set to ‘0’ by program because this flag cannot be cleared to ‘0’ by hardware.

(3) Timer Activation by Software

When starting the timer by software, set the TMnCNE flag to ‘1’ after initializing the timer. Setting the TMnCNE flag to ‘1’ starts the timer. Figure 5-9-1 shows the operation example. When starting the timer by external trigger, set the TMnCNE to ‘0’.

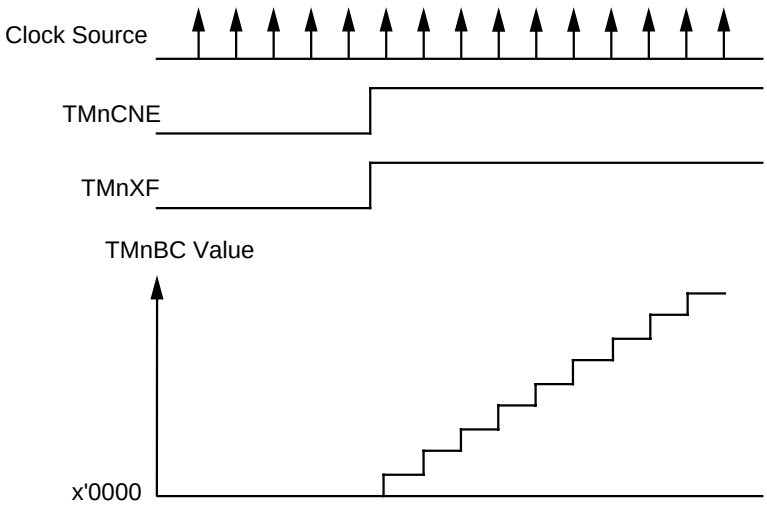


Figure 5-9-1 Timer Operation Example

(4) Up/Down Counter Select

Select up counting or down counting by setting the TMnUD[1:0] flags. Figure 5-9-2 and Figure 5-9-3 show the example of controlling up counting or down counting by the TMnIOA pin. The TMnUD[1:0] flags must be '00' when selecting a two-phase encoder as a clock source.

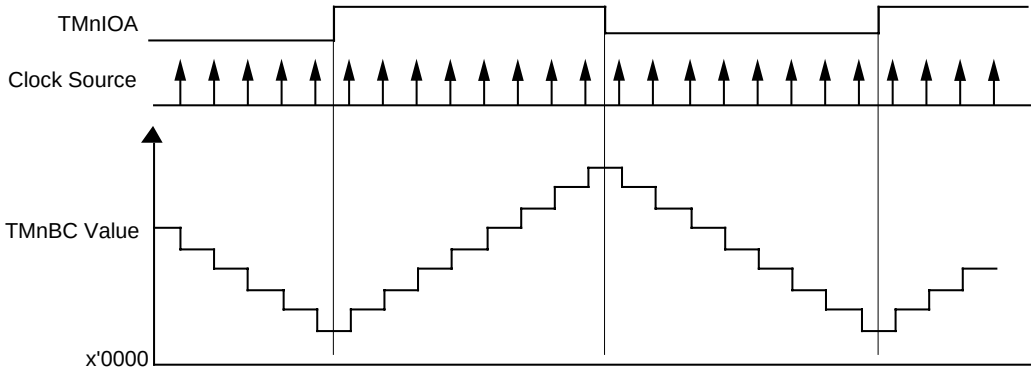


Figure 5-9-2 Up/Down Counting Select (TMnUD[1:0]='10')

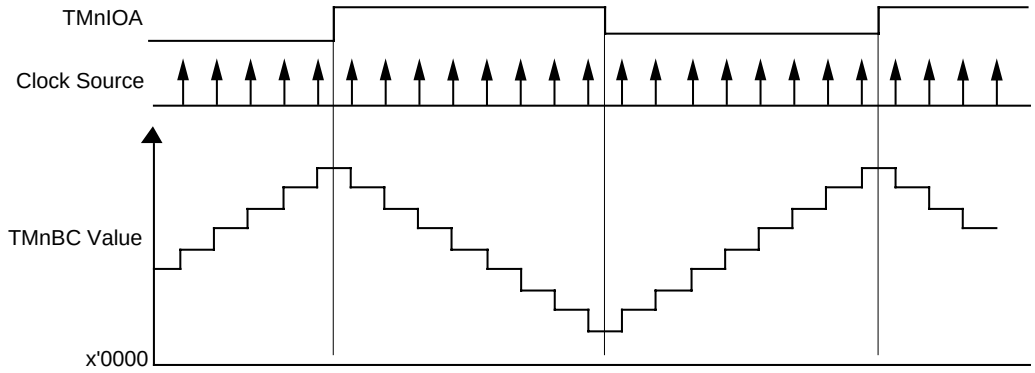


Figure 5-9-3 Up/Down Counting Select (TMnUD[1:0]='11')

(5) Counting Control Input

Set the counter input enable flag (TMnCGE) to '1' when controlling counter by TMnIOA pin. Mask the clock source of the binary counter based on the input level of the TMnIOA pin when TMnCGE is '1'. The input level of the TMnIOA pin for clock source masking is set by the TMnAEG flag. Figure 5-9-4 shows the example of the counting operation controlled by the TMnIOA pin.

Table 5-9-3 TMnIOA Input Level

	TMnAEG ="0"	TMnAEG ="1"
"H" Input	Normal Operation (Counting)	Clock Source Mask (Counting Stop)
"L" Input	Clock Source Mask (Counting Stop)	Normal Operation (Counting)

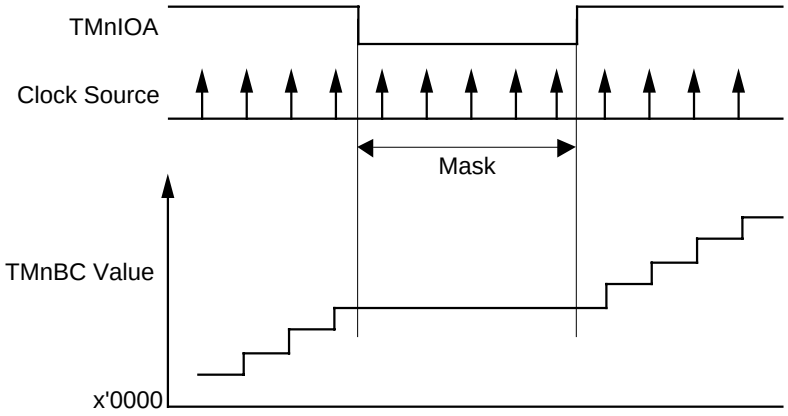


Figure 5-9-4 Count Operation Controlled by TMnIOA Pin

(6) Binary Counter Clear

Set the counter clear flag (TMnCLE) to '1' when clearing the binary counter by the TMnCA register. Clear the binary counter on the either condition of the following two conditions when TMnCLE is '1'.

Condition 1.
TMnCA is selected as a compare register.
TMnBC is a up counter.
TMnBC = TMnCA.

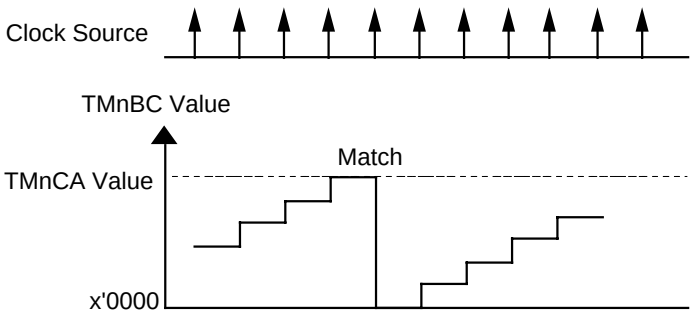


Figure 5-9-5 Operation Example 1

Condition 2.
Capture on TMnCA (TMnBC before clear is captured on TMnCA).

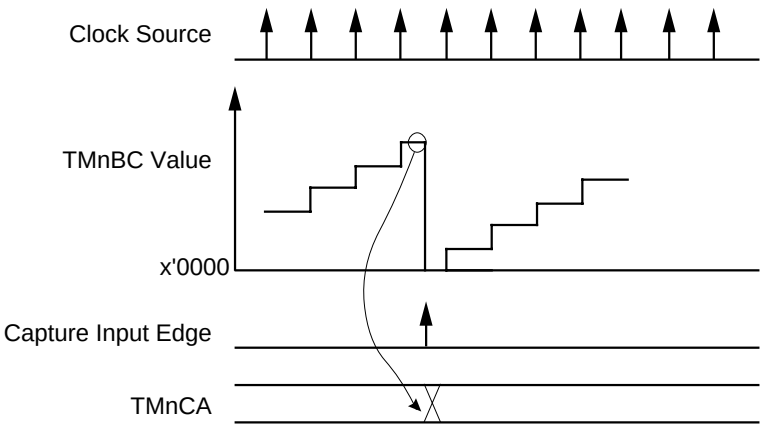


Figure 5-9-6 Operation Example 2

In addition, write the TMnCA value instead of x'FFFF' to the TMnBC counter when TMnBC overflows if TMnCLE is set to '1'.

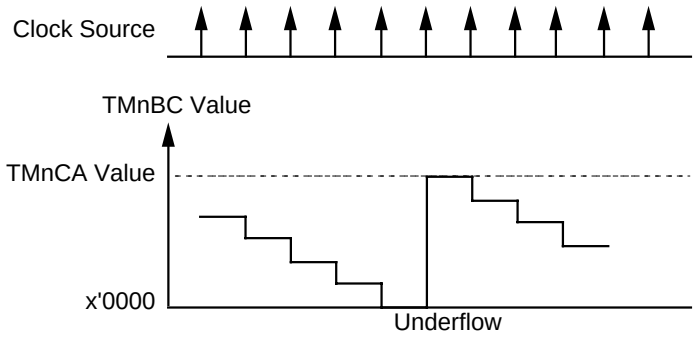


Figure 5-9-7 Operation Example 3

(7) Activation by External Trigger

Set the external trigger enable flag (TMnTGE) to '1' when activating the timer by TMnIOA trigger input. Timer starts when the selected edge is input to the TMnIOA pin if TMnTGE is set to '1'. The valid edge is selected by the TMnAEG flag. Setting TMnAEG to '0' selects the falling edge, while setting TMnAEG to '1' selects the rising edge.

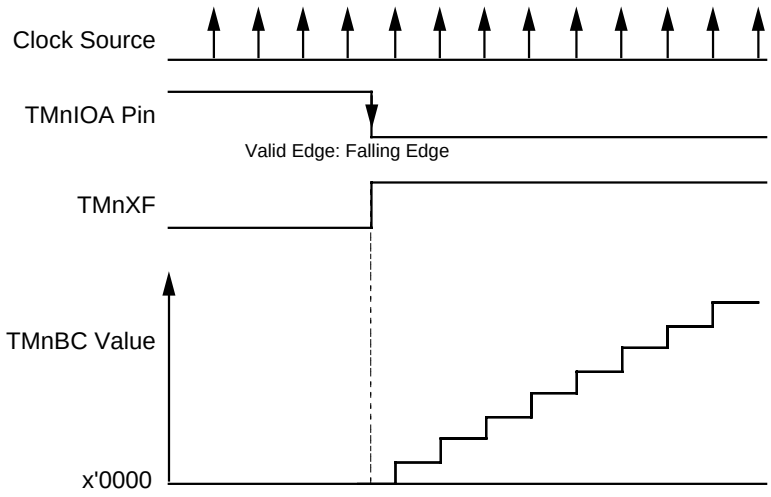


Figure 5-9-8 Operation Example 4

(8) One-shot Operation

Set the one-shot operation enable flag (TMnONE) to '1' when stopping the binary counter by the TMnCA register. Timer stops on the next clock cycle after TMnBC equals TMnCA if TMnONE is set to '1'. At this point, TMnBC equals TMnCA value plus 1 if TMnCLE is '0'. When the timer starts by setting the TMnCNE to '1', the TMnCNE remains '1' even though the timer stops. To activate the timer again, set the TMnCNE to '0' and then to '1'.

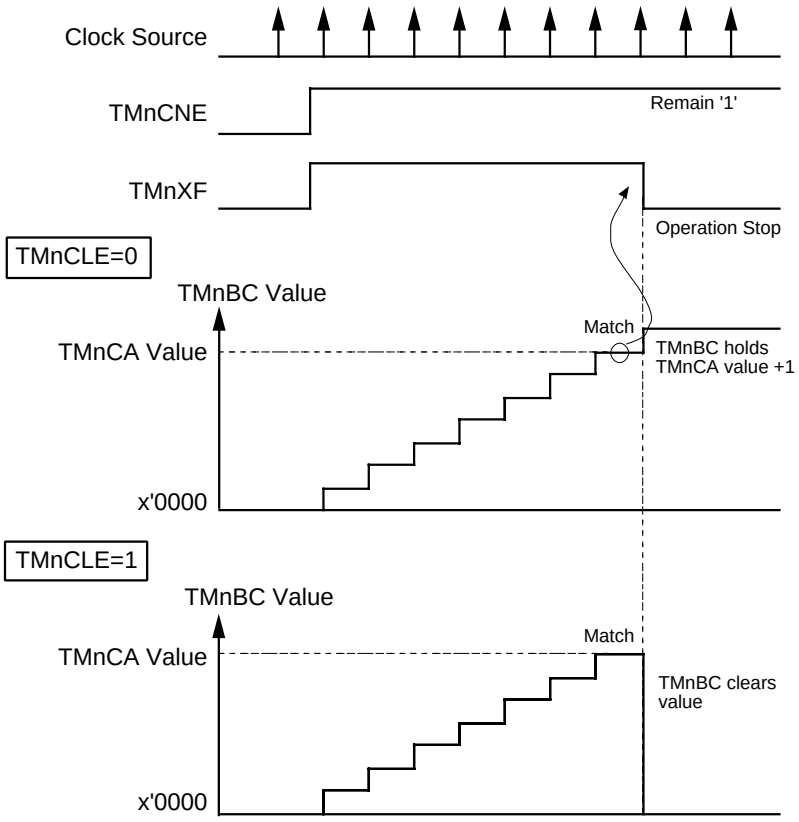


Figure 5-9-9 Operation Example 5

When the timer starts by the external trigger, the timer starts again if activation trigger is input to the TMnIOA pin even though the timer stops. (Do not need to set TMnTGE again.)

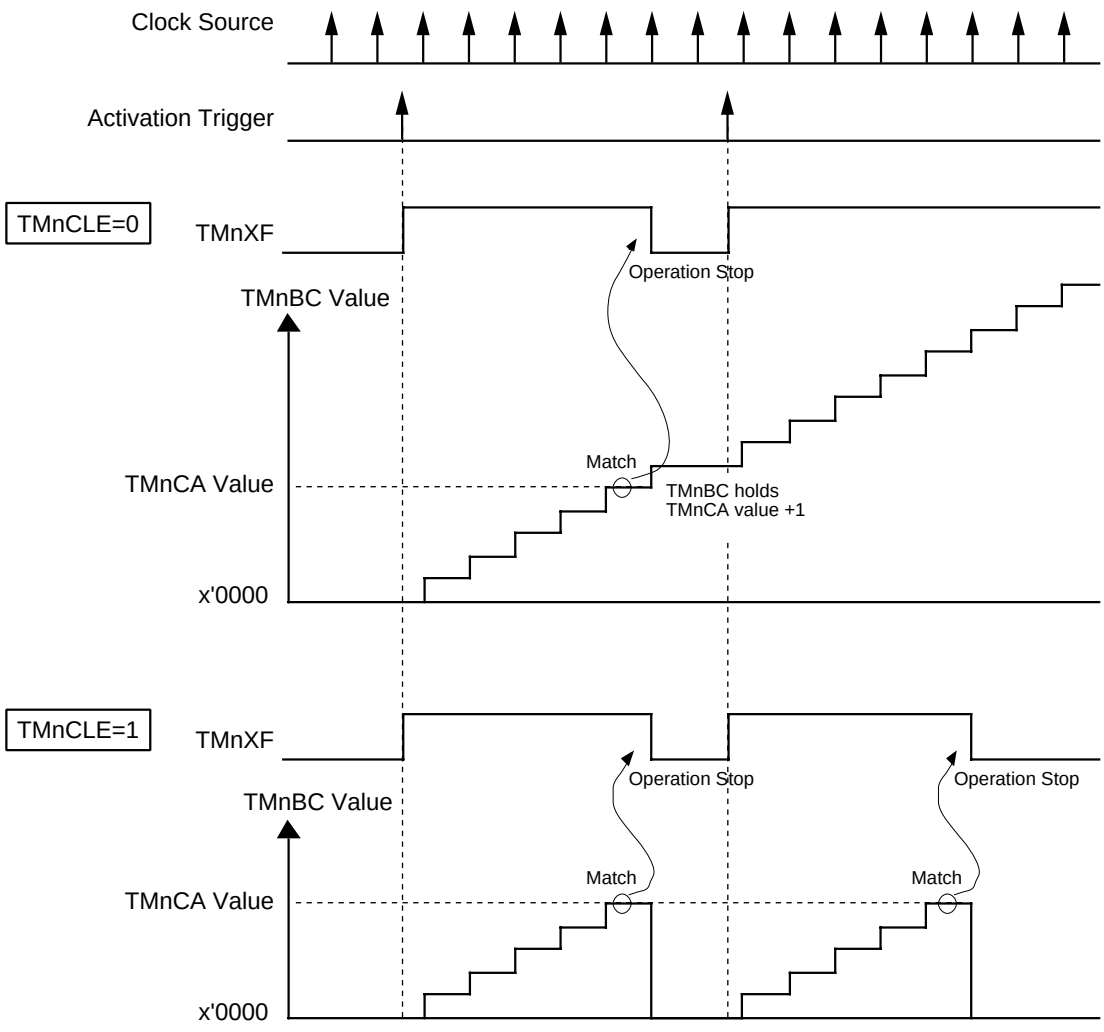


Figure 5-9-10 Operation Example 6

■ Timer Compare/Capture Register Setup

When the timer compare/capture A register (TMnCA) is controlled by the timer compare/capture A mode register (TMnMDA) , while the timer compare/capture B register (TMnCB) is controlled by the timer compare/capture B mode register (TMnMDB). Explain the timer compare/capture A mode register (the timer compare/capture B mode register) in this example.

(1) Compare Register/Capture Register Select

The timer compare/capture A operation mode select flag (TMnAM[1:0]) selects the TMnCA operating mode. The timer compare/capture B operation mode select flag (TMnBM[1:0]) selects the TMnCB operating mode.

Table 5-9-4 Compare/Capture Operating Mode Select

TMnAM[1:0] (TMnBM[1:0])	TMnCA(TMnCB) Operating Mode
00	Compare Register (Double Buffer)
01	Compare Register (Single Buffer)
10	Capture Register (Positive Edge/Negative Edge)
11	Capture Register (Both Edges)

The compare register mode has two modes of single buffer mode and double buffer mode. In the single buffer mode, the TMnCA (TMnCB) value becomes valid immediately after the value is written to the TMnCA(TMnCB) register. In the double buffer mode, the TMnCA (TMnCB) value is once latched to the compare register buffer and the value is not updated immediately after the value is written to the TMnCA (TMnCB) register. The compare register is updated on the following conditions.

1. TMnLDE = '1' (Timer Initialization)
2. TMnBC overflows or underflows while TMnCLE ='0'.
3. TMnBC counts up at TMnBC =TMnCA (TMnCB) while TMnCLE ='1' and TMnCA (TMnCB) is a compare register.

TMnCB is updated on the following condition as well as the above conditions.

4. Capture on TMnCB while TMnCLE ='1' and TMnCB is a capture register.

The capture register mode has two modes of either positive or negative edge mode and both edge mode. In either positive or negative mode, capture the TMnBC value on the TMnCA (TMnCB) when the selected edge is input to TMnIOA (TMnIOB) pin. In both edge mode, capture the TMnBC value on the TMnCA (TMnCB) regardless of the edges.

(2) Pin Polarity Select

TMnAEG and TMnBEG select the TMnIOA pin polarity and the TMnIOB pin polarity respectively.

Table 5-9-5 TMnIOA Pin Polarity Select

Function	TMnAEG = '0'	TMnAEG = '1'
Capture (Positive/Negative)	Rising Edge	Falling Edge
Count Control	Counting when TMnIOA is 1 Counting stops when TMnIOA is 0	Counting when TMnIOA is 0 Counting stops when TMnIOA is 1
Activation Trigger	Falling Edge	Rising Edge
Pin Output	Low at reset High at set	High at reset Low at set

Table 5-9-6 TMnIOB Pin Polarity Select

Function	TMnBEG = '0'	TMnBEG = '1'
Capture (Positive/Negative)	Rising Edge	Falling Edge
Clock Source Input (Positive/Negative)	Rising Edge	Falling Edge
Pin Output	Low at reset High at set	High at reset Low at set

(3) Capture Enable

TMnCA is enabled or disabled by the timer capture A operation enable flag (TMnACE) when TMnCA is selected as a capture register. TMnCB is enabled or disabled by the timer capture B operation enable flag (TMnBCE) when TMnCB is selected as a capture register.

Table 5-9-7 Capture Operating Mode Select

TMnACE(TMnBCE)	Function
0	Capture disable (Ignore edge input)
1	Capture enable

When TMnACE (TMnBCE) is '0', the TMnBC value is not captured on TMnCA (TMnCB). When the TMnBC value is captured on the TMnCA (TMnCB), TMnACE (TMnBCE) must be set to '1'. On the other hand, when TMnCA (TMnCB) is selected as a compare register, TMnACE (TMnBCE) must be set to '0'.

■ Pin Setup

The timer compare capture A mode register (TMnMDA) and the timer compare capture B mode register (TMnMDB) set the TMnIOA pin output waveform and the TMnIOB pin output waveform respectively.

(1) Output Waveform Polarity Select

The TMnAEG flag and the TMnBEG flag select the TMnIOA pin polarity and the TMnIOB pin polarity. The TMnIOA pin and the TMnIOB pin output the values set in TMnAEG and TMnBEG respectively when the timer is initialized (TMnLDE is '1'). When the TMnBC counter overflows/underflows or matches the TMnCA register/TMnCB register, the timer changes TMnIOA pin output and TMnIOB pin output based on the TMnAEG and TMnBEG respectively.

(2) Output Waveform Select

The timer A output waveform select flags (TMnAO[1:0]) and The timer B output waveform select flags (TMnBO[1:0]) select the TMnIOA pin output waveform and the TMnIOB pin output waveform respectively.

Table 5-9-8 TMnIOA Output Waveform Select

TMnAO[1:0]	Output Waveform to TMnIOA
00	Set the pin output controller when TMnBC matches TMnCA. Reset the pin output controller when TMnBC matches TMnCB.
01	Set the pin output controller when TMnBC matches TMnCA. Reset the pin output controller when TMnBC overflows.
10	Set the pin output controller when TMnBC matches TMnCA. (Reset when timer is initialized.)
11	Toggle Output (Output Invert when TMnBC matchesTMnCA.)

Table 5-9-9 TMnIOB Output Waveform Select

TMnBO[1:0]	Output Waveform to TMnIOB
00	Set the pin output controller when TMnBC matches TMnCB. Reset the pin output controller when TMnBC matches TMnCA.
01	Set the pin output controller when TMnBC matches TMnCB. Reset the pin output controller when TMnBC overflows.
10	Set the pin output controller when TMnBC matches TMnCB. (Reset when timer is initialized.)
11	Toggle Output (Output Invert when TMnBC matchesTMnCB.)

The pin output changes when TMnBC starts up counting or down counting at TMnBC = TMnCA (TMnCB) or TMnBC overflows.

The following shows the example of the TMnIOA (TMnIOB) pin output waveform.

The waveform to TMnIOA pin is “set when TMnBC = TMnCA, reset when TMnBC = TMnCB”. Reset is given priority over set if set and reset is generated at the same time.

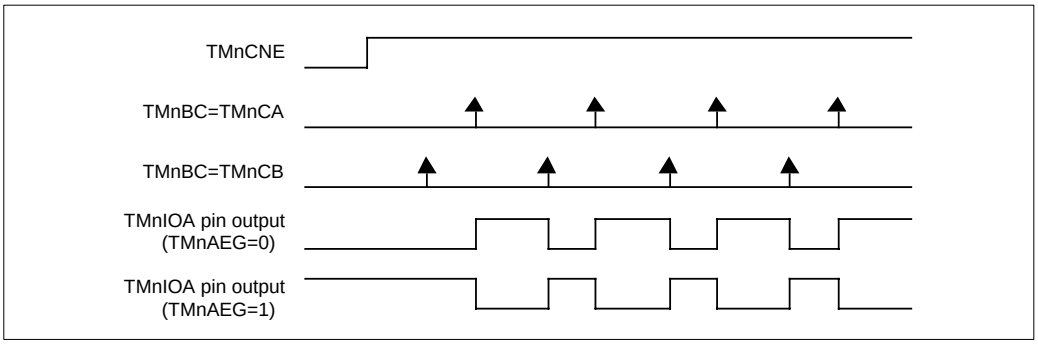


Figure 5-9-11 Pin Output Waveform 1

The waveform to TMnIOA pin is “set when TMnBC = TMnCA, reset when TMnBC overflows”. Reset is given priority over set if set and reset is generated at the same time.

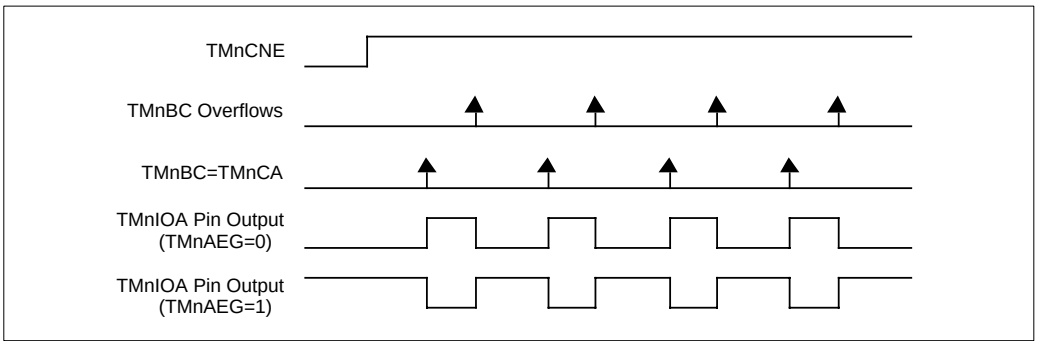


Figure 5-9-12 Pin Output Waveform 2

The waveform to TMnIOA pin is “set when TMnBC = TMnCA”.

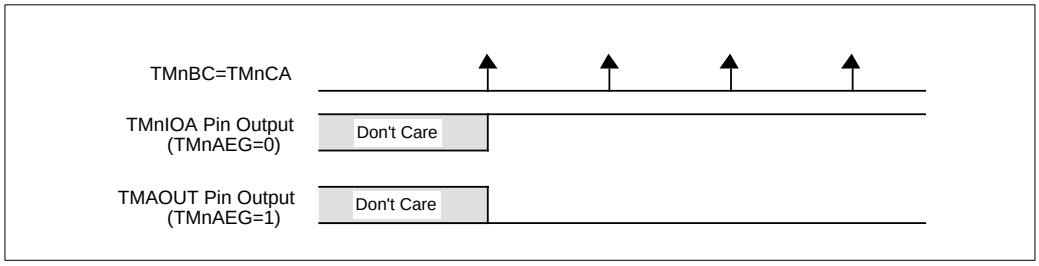


Figure 5-9-13 Pin Output Waveform 3

The waveform to TMnIOA pin is set “toggle output”.

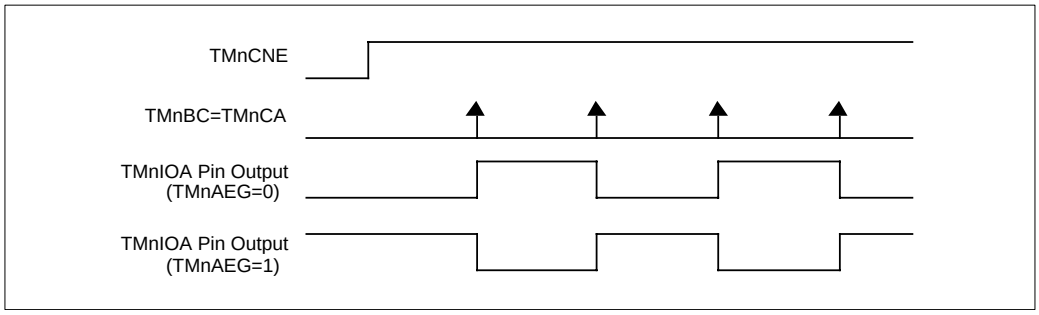


Figure 5-9-14 Pin Output Waveform 4

■ Interrupt Request

Timer generates 3 interrupt requests: over/underflow interrupt, compare capture A interrupt, compare capture B interrupt. Interrupt request signal is output during SYSCLK/1 cycle.

(1) Over/underflow Interrupt request

It is generated when TMnBC over/underflows.

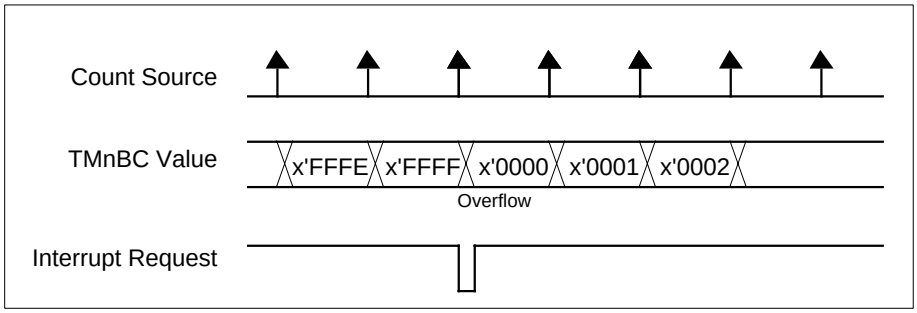


Figure 5-9-15 At Upcount

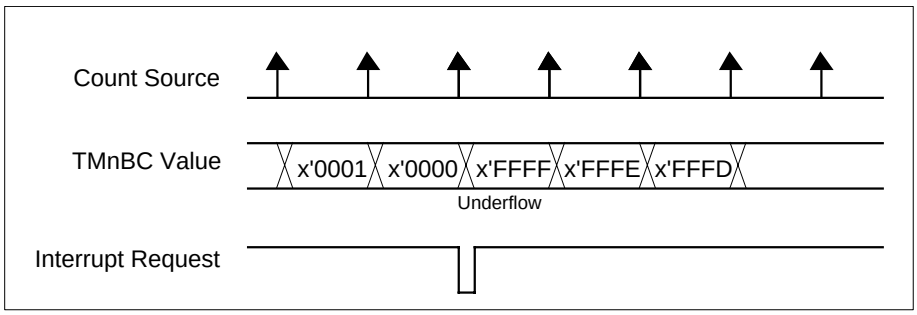


Figure 5-9-16 At Downcount

Underflow interrupt is also generated when TMnCLE='1'.

(2) Compare Capture A (B) Interrupt Request

When TMnCA (TMnCB) is set to the capture register, interrupt request is generated at capturing. When it is set to the compare register, interrupt request is generated at up/downcount from the status matched with TMnBC.

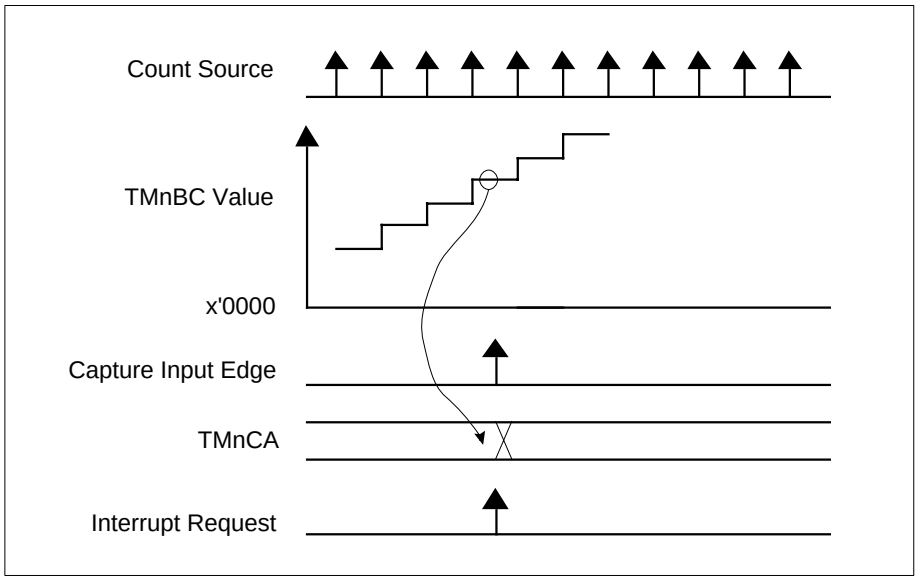


Figure 5-9-17 Compare Capture A (B) Interrupt Request 1

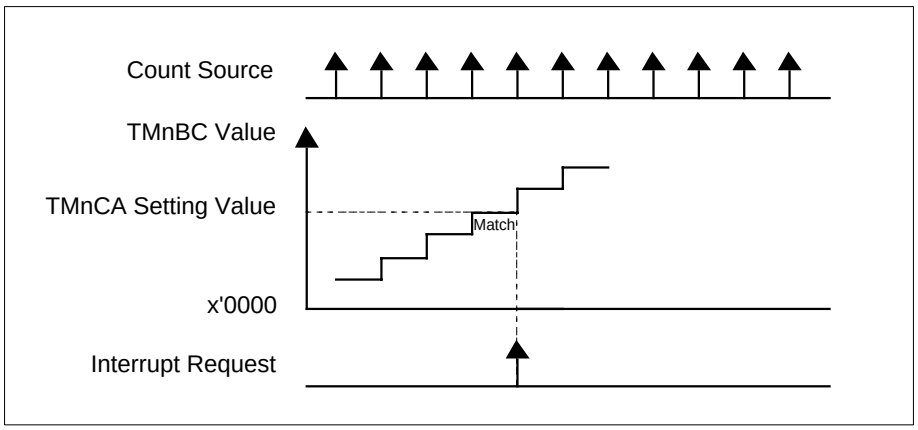


Figure 5-9-18 Compare Capture A (B) Interrupt Request 2

5-9-2 Interval Timer Operation Examples

This section explains interval timer operations. Following shows the setting procedure when timer is used as interval timer. It operates as interval timer which generates compare capture A interrupt by set cycle. (Figure 5-9-19 to 5-9-22) Compare capture register B register is able to use as compare register or capture register.

■ Operation Start Procedure

(1) Set the mode of compare capture register A.
Set TMnMDA register as follows
TMnAO1, 0 Arbitrary
TMnACE 0 : Disable capture operation
TMnAEG Arbitrary
TMnAM1, 0 00 : Compare register (single buffer)
or
01 : Compare register (double buffer)
When interrupt cycle is changed during count operation, set to double buffer.

(2) Set division rate of timer
Set division rate to TMnCA.
Compare capture A interrupt cycle is:
(TMnCA setting value + 1) × count source cycle

(3) Set operation mode.
Set TMnMD register as follows
TMnCK2, 1, 0 Arbitrary : Select count source
TMnLDE 0 : Normal operation
TMnCNE 0 : Stop count operation
TMnUD1,0 00 : Upcount
TMnCGE 0 : Disable to control count by TMnIOA pin input
TMnCLE 1 : Clear TMnBC when TMnCA = TMnBC
TMnONE 0 : Disable one-shot operation
TMnTGE 0 : Disable to start timer by external trigger
When 1/8 SYSCLK is used as count source, before timer count operation is enabled, set TMPSnCNE of prescaler control register (PSCNTn) to '1', and enable prescaler operation.

(4) Initialize timer.
Initialize timer by setting TMnLDE of TMnMD register to ‘1’. TMnBC is cleared and pin output is reset. Also, if TMnCA is set as compare register of double buffer, setting value is loaded from buffer to compare register. After initializing, set TMnLDE to ‘0’ and return to normal operation.

(5) Enable timer count operation
When TMnCNE of TMnMD register is set ‘1’, TMnXF becomes ‘1’ and timer starts operation.

It generates compare capture A interrupt by fixed cycle, and operates as interval timer. If TMnCA register value is changed during count operation, when TMnBC is cleared next time, it is loaded from buffer to compare register and interrupt cycle is changed. For that, set TMnCA to compare register of double buffer.

■ Operation Stop Procedure

(1) Stop timer count operation.
When TMnCNE of TMnMD register is set ‘0’, TMnXF becomes ‘0’ and timer stops operation.

(2) Initialize timer according to need.
When TMnLDE of TMnMD register is set ‘1’, TMnBC is cleared and timer output is reset. Also, if TMnCA is set as double buffer, setting value is loaded from buffer to compare register.
After timer stops, if TMnLDE is not set to ‘1’, binary counter, compare register and pin output maintains the status before stop. If TMnCNE is set to ‘1’ again, count is started again from the status just before stop.

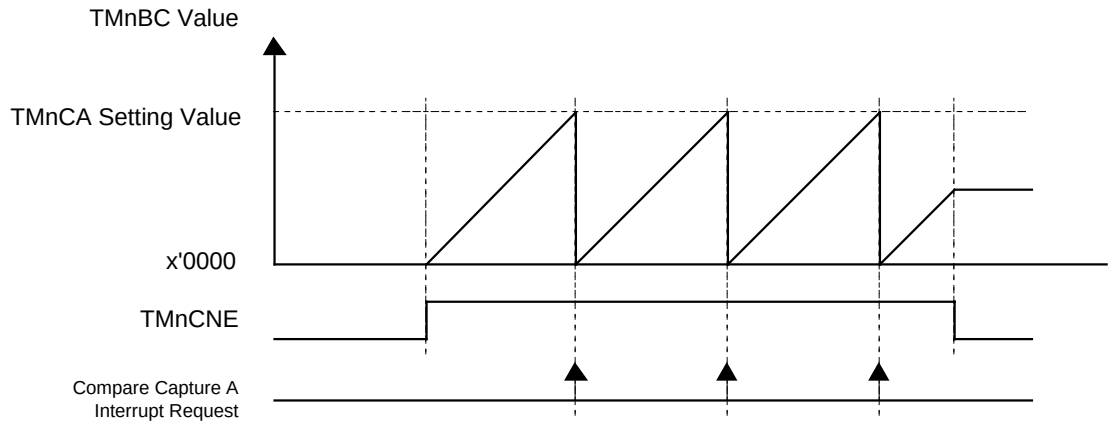


Figure 5-9-19 Interval Timer Operation 1

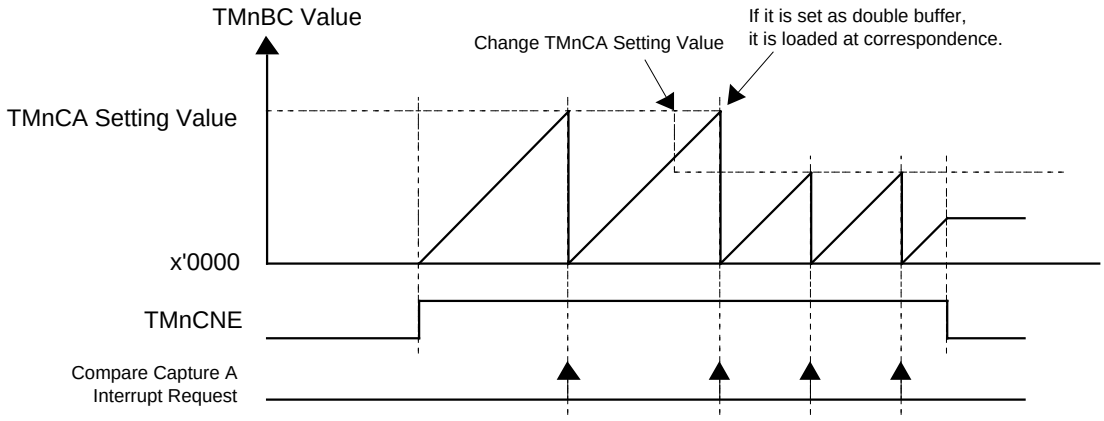


Figure 5-9-20 Interval Timer Operation 2

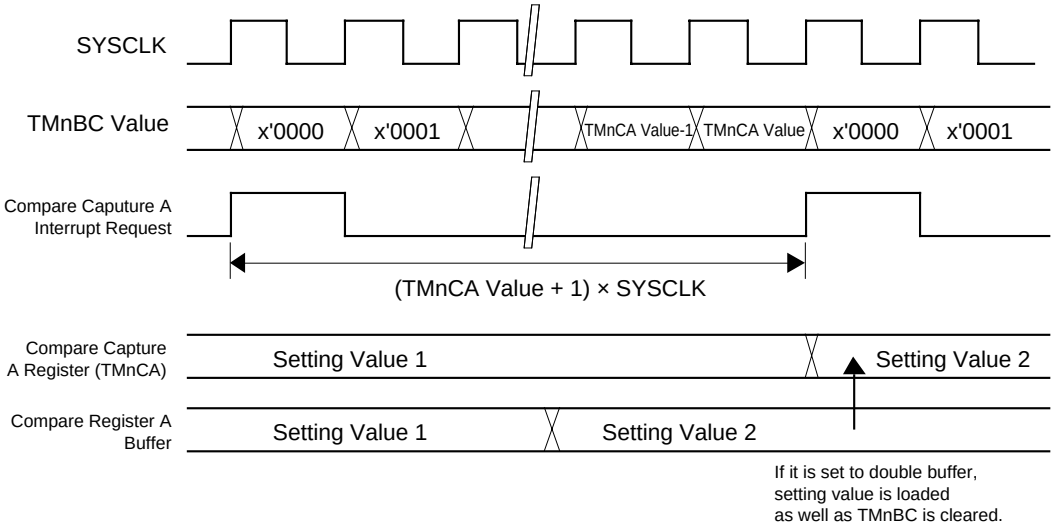


Figure 5-9-21 Interval Timer Operation (Count Source = SYSCLK)

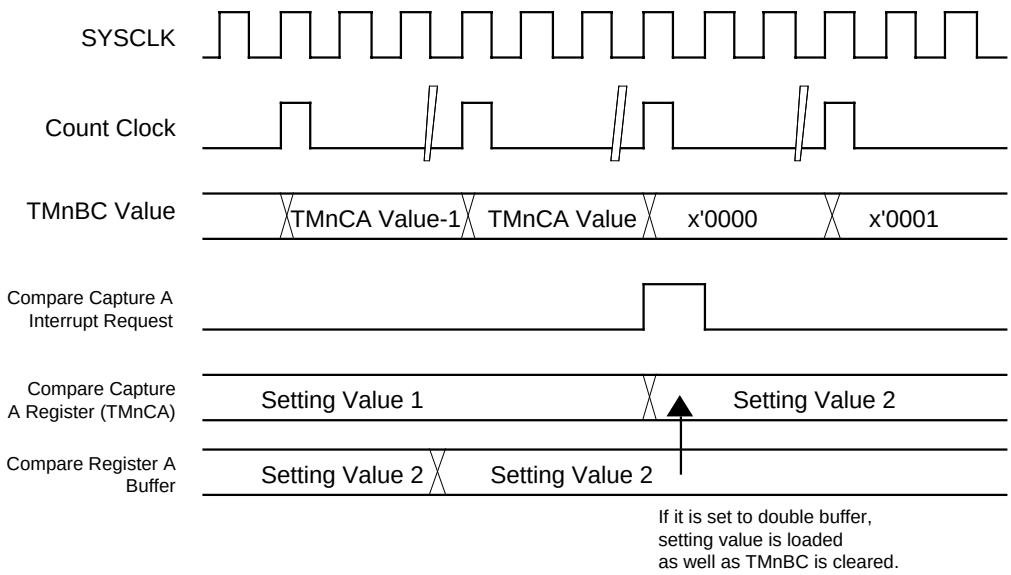


Figure 5-9-22 Interval Timer Operation (Prescaler Used)

Chapter 6 Serial Interface

6-1 Serial Interface

The MN102H74G/74F/74D/F74G contains four serial interfaces that can be used for synchronous mode and UART mode.

6-1-1 Serial Interface Function

The following table shows functions of each serial interface.

Table 6-1-1 Serial Interface 0,1 Function

Communication	Clock Synchronous Mode	UART (Asynchronous) Mode	I^2C Mode
Parity	None, 0, 1 Odd, Even		Master transmission and reception are possible. (No start sequence detection function)
Character Length	7 bits, 8 bits		
Bit Order	LSB first, MSB first		
Clock Source	Serial 0: 1/2 and 1/16 of Timer 6 underflow, SBT0 Serial 1: 1/2 and 1/16 of Timer 7 underflow, SBT1	Serial 0: 1/16 of Timer 6 underflow Serial 1: 1/16 of Timer 7 underflow	
Maximum Baud Rate	3 Mbps	375 kbps	
Error Detection	Parity error, Overrun error	Parity error, Overrun error, Framing error	
Buffer	Independent transmit/receive buffer (double transmit buffer, double receive buffer)		
Interrupt	Transmit buffer empty/transmission end interrupt, reception end interrupt		

Table 6-1-2 Serial Interface 2,3 Function

Communication	Clock Synchronous Mode	UART (Asynchronous) Mode
Parity	None, 0, 1 Odd, Even	
Character Length	7 bits, 8 bits	
Bit Order	LSB first, MSB first	
Clock Source	Serial 2: 1/2 and 1/16 of Timer 8 underflow, SBT2 Serial 3: 1/2 and 1/16 of Timer 9 underflow, SBT3	Serial 2: 1/16 of Timer 8 underflow Serial 3: 1/16 of Timer 9 underflow
Maximum Baud Rate	3 Mbps (*Note)	375 kbps
Error Detection	Parity error, Overrun error	Parity error, Overrun error, Framing error
Buffer	Independent transmit/receive buffer (single transmit buffer, double receive buffer)	
Interrupt	transmission end interrupt, reception end interrupt	

Note) For sequence synchronous reception in serial interface 2, 3 using external clock (SBTn : n =2,3) at transmission rate over 1.5 Mbps, keep an interval at least 4 machine cycles (at app.333 ns, 12 MHz oscillator) between data receptions.

6-1-2 Serial Interface Block Diagram

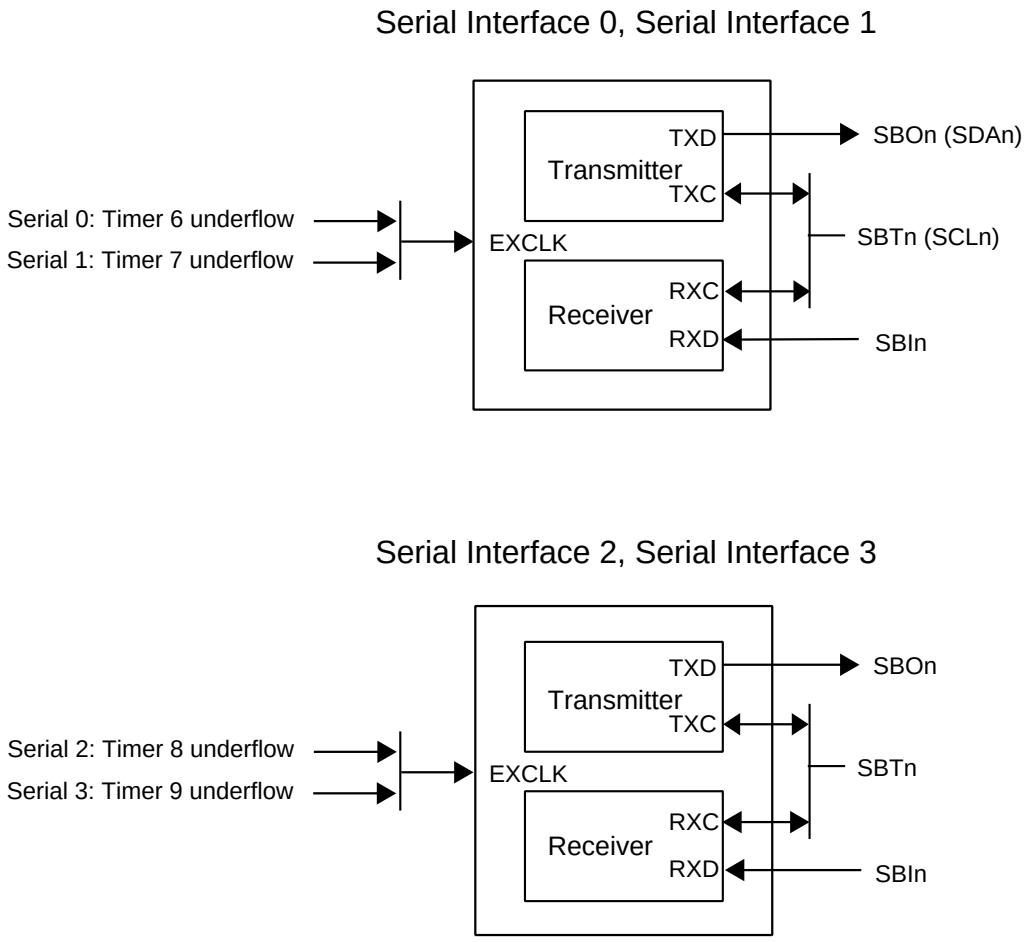


Figure 6-1-1 Serial Interface Block Diagram

6-2 Control Registers

6-2-1 List of Serial Interface Control Registers

The following table shows registers to control serial interface.

Table 6-2-1 List of Serial Interface Control Registers

Register		Address	R/W	Function
Serial 0	SCA0CTR	x'E00200'	R/W	Serial 0 Control Register
	SCA0RB	x'E00202'	R	Serial 0 Receive Buffer
	SCA0STR	x'E00203'	R	Serial 0 Status Register
	SCA0TB	x'E00204'	R/W	Serial 0 Tramsmit Buffer
	SCA0STR1	x'E00206'	R	Serial 0 Status 1 Register
Serial 1	SCA1CTR	x'E00210'	R/W	Serial 1 Control Register
	SCA1RB	x'E00212'	R	Serial 1 Receive Buffer
	SCA1STR	x'E00213'	R	Serial 1 Status Register
	SCA1TB	x'E00214'	R/W	Serial 1 Transmit Buffer
	SCA1STR1	x'E00216'	R	Serial 1 Status 1 Register
Serial 2	SCA2CTR	x'E00220'	R/W	Serial 2 Control Register
	SCA2RB	x'E00222'	R	Serial 2 Receive Buffer
	SCA2STR	x'E00223'	R	Serial 2 Status Register
	SCA2TB	x'E00224'	R/W	Serial 2 Transmit Buffer
Serial3	SCA3CTR	x'E00230'	R/W	Serial 3 Control Register
	SCA3RB	x'E00232'	R	Serial 3 Receive Buffer
	SCA3STR	x'E00233'	R	Serial 3 Status Register
	SCA3TB	x'E00234'	R/W	Serial 3 Transmit Buffer

6-2-2 Serial Reception Registers/Serial Transmission Registers

■ Serial n Reception Registers (SCAnRB n=0 to 3)

The data is read from the SCAnRB register during the serial reception. The data is received when an interrupt occurs or the SCAnRXA flag of the SCAnSTR register is 1. The MSB (bit 7) becomes 0 in 7-bit transfer. The serial interface operations cannot be guaranteed when this register is written. The receive busy flag (SCAnRBSY) of the SCAnSTR register is cleared when this register is read in 16-bit access.

	7	6	5	4	3	2	1	0
	SCAn RB7	SCAn RB6	SCAn RB5	SCAn RB4	SCAn RB3	SCAn RB2	SCAn RB1	SCAn RB0
Reset	0	0	0	0	0	0	0	0

Figure 6-2-1 Serial Reception Register (SCAnRB, R)
(n=0: x'E00202', n=1: x'E00212', n=2: x'E00222', n=3: x'E00232')

MSB (bit 7) becomes 0 during 7-bit data transfer.

■ Serial n Transmission Registers (SCAnTB n=0 to 3)

The data transmission starts by writing the serial transmission register (SCAnTB). The transmission starts in 4 cycles of clock for transmission from writing SCAnTB. The MSB (bit 7) is ignored in 7-bit transfer.

	7	6	5	4	3	2	1	0
	SCAn TB7	SCAn TB6	SCAn TB5	SCAn TB4	SCAn TB3	SCAn TB2	SCAn TB1	SCAn TB0
Reset	0	0	0	0	0	0	0	0

Figure 6-2-2 Serial Transmission Register (SCAnTB, R/W)
(n=0: x'E00204', n=1: x'E00214', n=2: x'E00224', n=3: x'E00234')

MSB (bit 7) is ignored during 7-bit data transfer.

6-2-3 Serial Control Registers

■ Serial n Control Registers (SCAnCTR n=0 to 3)

The serial n control register (SCAnCTR) sets the conditions to control the serial interface operations.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCAn TEN	SCAn REN	SCAn BRE	SCAn IIC	SCAn PTL	SCAn TI	SCAn OD	SCAn ICM	SCAn LN	SCAn PTY2	SCAn PTY1	SCAn PTY0	SCAn SB	—	SCAn S1	SCAn S0
Reset 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCAn S1	SCAn S0	Clock Source
0	0	SBTn
	1	Serial 0: Timer 6 underflow/2 Serial 1: Timer 7 underflow/2 Serial 2: Timer 8 underflow/2 Serial 3: Timer 9 underflow/2
1	0	Reserved
	1	Serial 0: Timer 6 underflow/16 Serial 1: Timer 7 underflow/16 Serial 2: Timer 8 underflow/16 Serial 3: Timer 9 underflow/16

Timer n underflow/2 cannot be selected in asynchronous mode.

SCAnSB	Stop Bit
0	1 bit
1	2 bits

Select stop bit only in asynchronous mode.

SCAn PTY2	SCAn PTY1	SCAn PTY0	Parity Bit
0	0	0	No parity
		1	Reserved
	1	0	Reserved
		1	Reserved
1	0	0	0 (output low)
		1	1 (output high)
	1	0	Even (1s are even)
		1	Odd (1s are odd)

SCAnLN	Character Length
0	7 bits
1	8 bits

LSB first is selected in 7-bit transfer.

SCAnICM	I^2C Mode
0	Off
1	On

Set to 0 for serial interface 2 and serial interface 3.

SCAnOD	Bit Order
0	LSB first
1	MSB first

SCAnTI	Transmission End Interrupt
0	Transmission end interrupt
1	Transmission buffer empty

Set to 0 for serial interface 2 and serial interface 3.

SCAnPTL	Protocol
0	Asynchronous
1	Synchronous/I^2C

SCAnIIC	I^2C Mode Select
0	Stop sequence output when changing from 1 to 0
1	Start sequence output when changing from 0 to 1

Set to 0 for serial interface 2 and serial interface 3.

SCAnBRE	Break Transmission
0	Do not break
1	Break

SCAnREN	Receive Enable
0	Disable
1	Enable

SCAnTEN	Transmit Enable
0	Disable
1	Enable

Figure 6-2-3 Serial Control Register (SCAnCTR, R/W)
(n=0: x'E00200', n=1: x'E00210', n=2: x'E00220', n=3: x'E00230')

6-2-4 Serial Interface Status Registers

■ Serial n Status Register (SCAnSTR n=0 to 3)

The serial interface operations are not guaranteed if the serial status register (SCAnSTR) is written. A framing error occurs when the stop bit is 0. The framing error data is updated whenever the stop bit is received. A parity error occurs when the parity bit is 1 although it is set to 0, it is 0 although it is set to 1, it is odd although it is set to even and it is even although it is set to odd. The parity error data is updated whenever the parity bit is received. An overrun error occurs when the next data is received completely before the received data (SCAnRB) is read. The overrun error data is updated whenever the last bit of data is received.

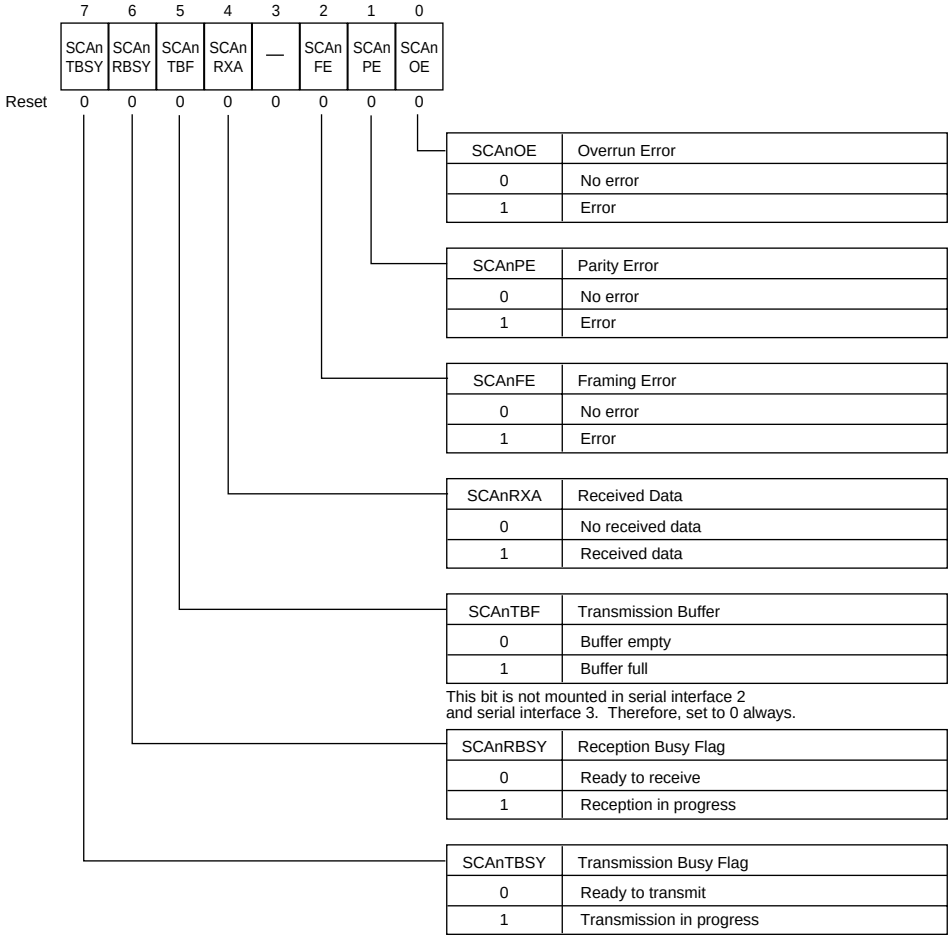


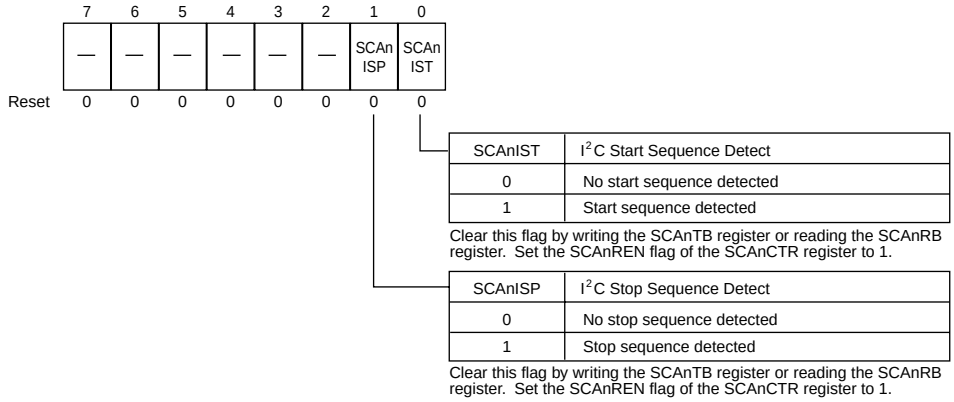
Figure 6-2-4 Serial Status Register (SCAnSTR, R)
(n=0: x'E00203', n=1: x'E00213', n=2: x'E00223', n=3: x'E00233')

Do not write the serial n status registers.

6-2-5 Serial Status 1 Register

■ Serial n Status 1 Register (SCAnSTR1 n=0, 1)

The serial interface operations are not guaranteed if the serial status 1 register (SCAnSTR1) is written.



SCAnIST	I^2C Start Sequence Detect
0	No start sequence detected
1	Start sequence detected

Clear this flag by writing the SCAnTB register or reading the SCAnRB register. Set the SCAnREN flag of the SCAnCTR register to 1.

SCAnISP	I^2C Stop Sequence Detect
0	No stop sequence detected
1	Stop sequence detected

Clear this flag by writing the SCAnTB register or reading the SCAnRB register. Set the SCAnREN flag of the SCAnCTR register to 1.

Figure 6-2-5 Serial Status 1 Register (SCAnSTR1, R)
(n=0: x'E00206', n=1: x'E00216')

Do not write the serial status registers.

6-3 Serial Interface Operation

6-3-1 Serial Interface Operation

■ Serial Interface Connection

This section describes the serial interface connection examples.

Asynchronous Connection

The serial interface can connect using simplex or duplex transfer mode.

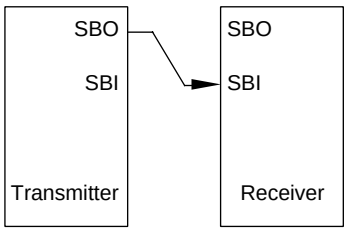


Figure 6-3-1 Asynchronous Simplex Mode

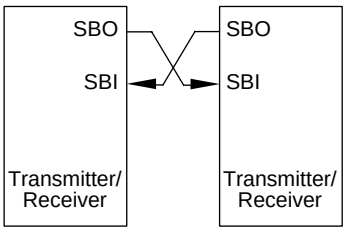


Figure 6-3-2 Asynchronous Duplex Mode

Synchronous Connection

The serial interface can connect using simplex or duplex transfer mode.

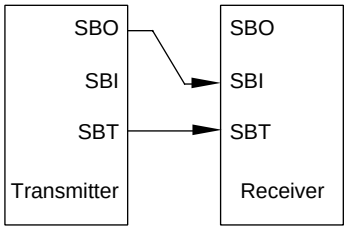


Figure 6-3-3 Synchronous Simplex Mode

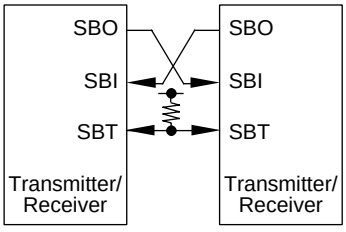


Figure 6-3-4 Synchronous Duplex Mode

I^2C Mode (only for serial interface 0 and serial interface 1)

The serial interface can connect to devices with slave transmitter/receivers.

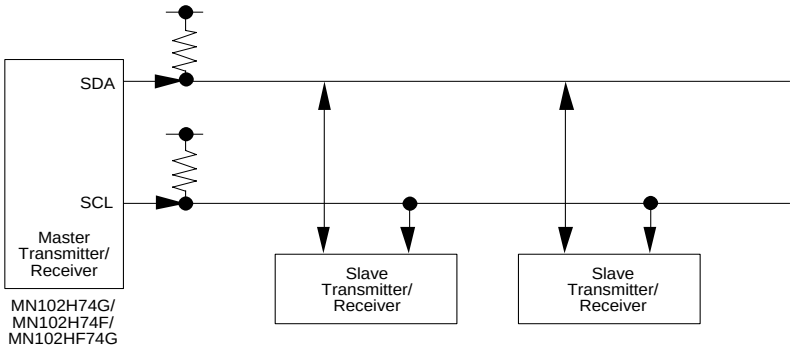


Figure 6-3-5 I^2C Mode Connection

The SDA and SCL pins require pull-up resistors in duplex transfer because these pins are input pins if both are not transmission.

■ Clock Selection

The transfer clock for serial interface in asynchronous mode is set to timer underflow/16. On the other hand, the transfer clock in synchronous mode is set to timer underflow/16 or timer underflow/2. The following example shows timer 6 underflow setup as clock for serial interface 0. Timer 6 underflow outputs the system clock (12 MHz) divided by timer 6.

Table 6-3-1 Baud Rate Setup in Asynchronous Mode

Baud Rate (bps)	Division of Timer 6
31250	24
28800	26
19200	39
14400	52
9600	78
4800	156
2400	312
1200	624

The value set to the timer base register is calculated as follow.

Underflow Cycle = (Timer base register value +1) × System clock cycle

Baud Rate = 1/(underflow cycle × 16)

“16” is the divisor of clock source.

Based on two above formula,

Timer base register value = System Clock/(Baud Rate × 16) -1

■ Asynchronous Timing Chart

8-bit character length, no parity, 2 stop bits

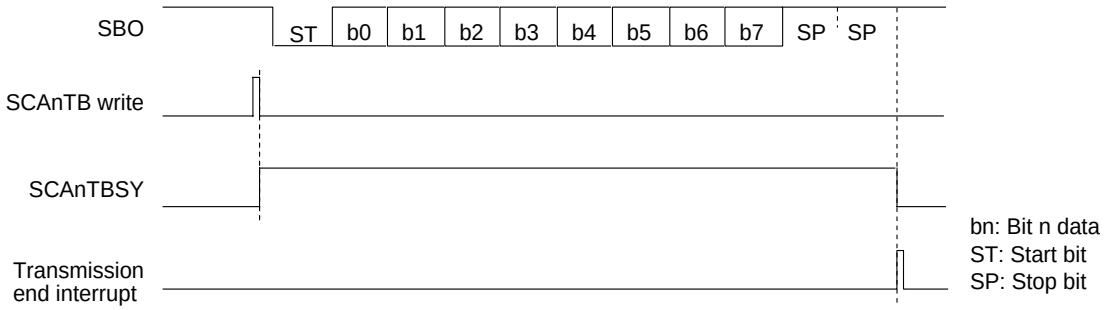


Figure 6-3-6 Asynchronous Transmission Timing

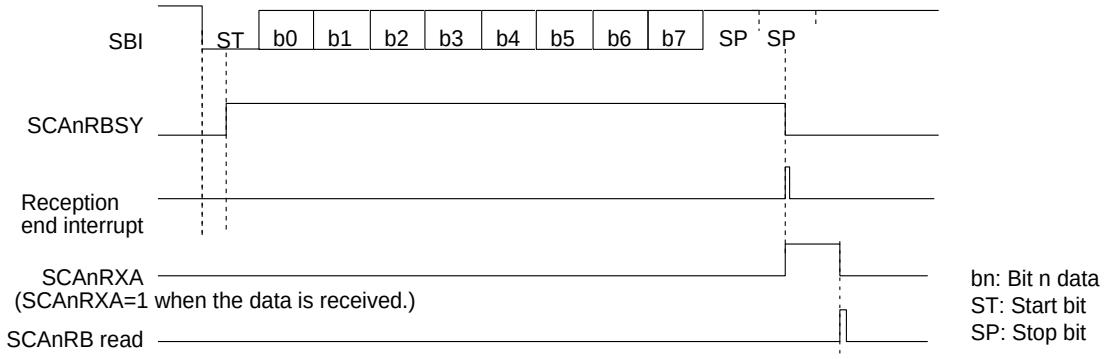


Figure 6-3-7 Asynchronous Reception Timing

■ Synchronous Timing Chart

8-bit character length, 1 parity bit

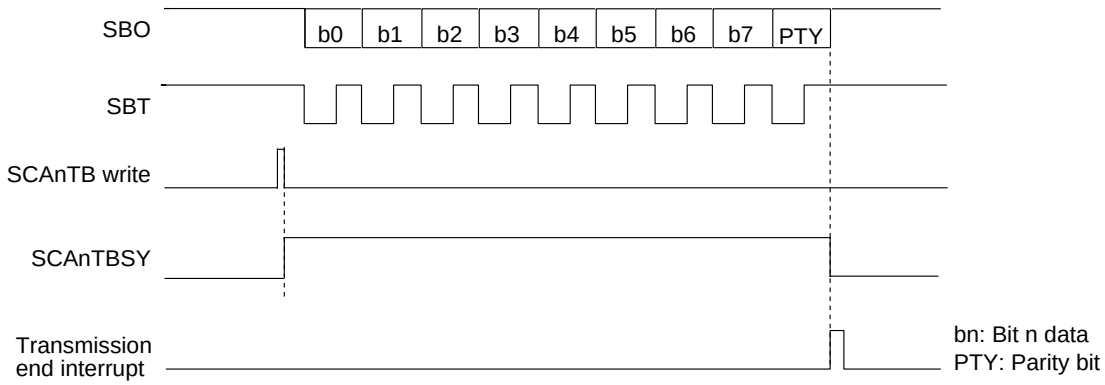


Figure 6-3-8 Synchronous Transmission Timing

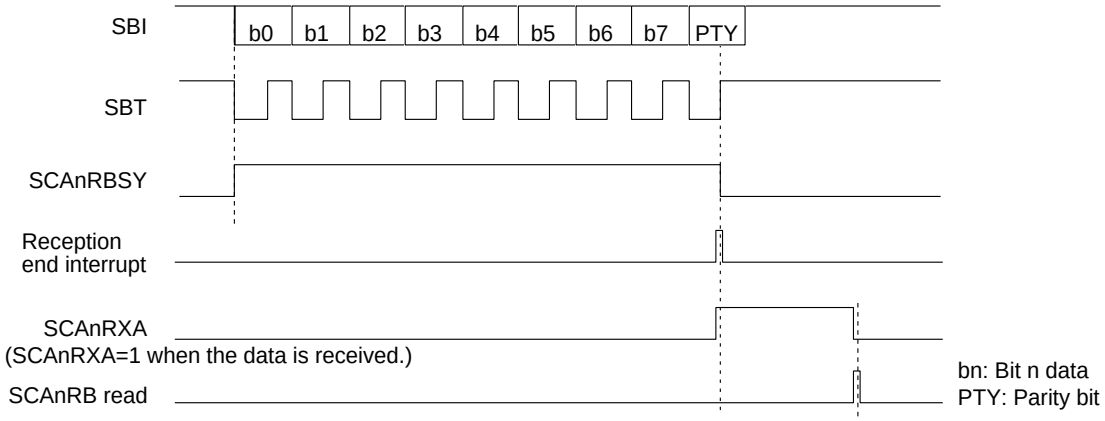


Figure 6-3-9 Synchronous Reception Timing

6-3-2 Serial Interface 0 Setup Example

This section describes the example of serial transmission in asynchronous mode with the following setting.

- Transfer Rate = 2400 bps
- LSB first order bit
- 8-bit data transfer
- Two stop bits
- Odd parity

The next data is transmitted in a serial transmission end interrupt service routine when a serial transmission end interrupt occurs.

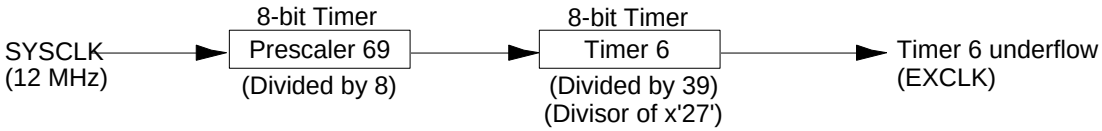


Figure 6-3-10 Asynchronous Transmission Configuration

The serial transmission starts by writing the data to the SCA0TB register. The serial transmission starts synchronizing with timer 6 underflow. The next data is transmitted in a serial transmission end interrupt service routine when a serial transmission end interrupt occurs. When an interrupt is not used, the CPU needs polling the SCAnTBSY flag of the SCA0STR register.

Setting the timer is required in asynchronous reception mode.

The serial interface generates the serial transfer baud rate on the timer 6 underflow/16. In this example, prescaler 69 divides SYSCLK by 8 and timer 6 divides SYSCLK/8 by 39.

■ Prescaler 69 Setup

(1) Verify that counting has stopped by the prescaler 69 control register (PSCNT69).

This step is not required immediately after reset.

PSCNT69: x'E00004'

	7	6	5	4	3	2	1	0
	TMPS CNE	-	-	-	-	-	-	-
Reset	0							

(2) Start prescaler 69 by setting TMPSCNE to '1'.

■ Timer 6 Setup

(3) Verify that counting has stopped by the timer 6 mode register (TM6MD).

This step is not required immediately after reset.

TM6MD: x'E00034'

	7	6	5	4	3	2	1	0
	TM6 CNE	TM6 LDE	-	-	-	TM6 CK2	TM6 CK1	TM6 CK0
Reset	0	0	-	-	-	0	0	0

(4) Set the timer 6 divisor. Since the divisor is the prescaler 69 output/39, set the timer 6 base register (TM6BR) to 38. (The valid range for TM6BR is 1 to 255.)

TM6BR: x'E00024'

7	6	5	4	3	2	1	0
TM6 BR7	TM6 BR6	TM6 BR5	TM6 BR4	TM6 BR3	TM6 BR2	TM6 BR1	TM6 BR0
0	0	1	0	0	1	1	0

(5) Set the TM6LDE flag and the TM6CNE flag to 1 and 0 respectively. At the same time, set the TM6CKn flags to 001 to select the clock source.

Do not change the clock source once you have selected. Changing the clock source while setting the count operation control will corrupt the value in the binary counter.

(6) Set both the TM6LDE flag and the TM6CNE flag to 0.

If this setting is omitted, TM0BC does not count at the first count.

(7) Set the TM6LDE flag and the TM6CNE flag to 0 and 1 respectively. This starts timer 6.

Counting starts at the start of the next cycle.

■ Serial Interface 0 Setup

(8) Enable interrupts after clearing all existing interrupt requests. To do this, set the IRQ5LV[2:0] flags, NIRQ9IR and NIRQ9IE of the maskable interrupt control register (G5ICR) to 0 and 1 respectively. (for example, set G5ICR to x'4200'.) Set the SC0TIE flag of the subgroup maskable interrupt control register 50 (SG50ICR) to 1. Thereafter, a serial transmission end interrupt occurs when the data transmission ends.

G5ICR: x'00FC4A'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IRQ5 LV2	IRQ5 LV1	IRQ5 LV0	-	-	NIRQ9 IE	NIRQ8 IE	-	-	NIRQ9 IR	NIRQ8 IR	-	-	NIRQ9 ID	NIRQ8 ID
-	1	0	0	-	-	1	0	-	-	0	0	-	-	0	0

SG50ICR: x'E00602'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	SC0T IE	SC0R IE	TM3 IE	TM2 IE	SC0T IR	SC0R IR	TM3 IR	TM2 IR	SC0T ID	SC0R ID	TM3 ID	TM2 ID
-	-	-	-	0	1	0	0	0	0	0	0	0	0	0	0

(9) Set the serial interface operating conditions to the serial control register (SCA0CTR). Select LSB first bit order, timer 6 underflow/16, 8-bit data transfer, 2 stop bits and odd parity.

SCA0CTR: x'E00200'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
0	0	0	0	0	0	0	0	1	1	1	1	1	-	1	1

(10) Set either SCA0TEN or SCA0REN, or both SCA0TEN and SCA0RENof the SCA0CTR register.

SCA0CTR: x'E00200'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
1	1	0	0	0	0	0	0	1	1	1	1	1	-	1	1

(11) Set the first transfer data to the serial transmission register (SCA0TB). When the transfer data is set to the SCA0TB register, the serial transmission starts synchronizing with timer 6. If an interrupt occurs, the interrupt service routine and the next data transfer start.

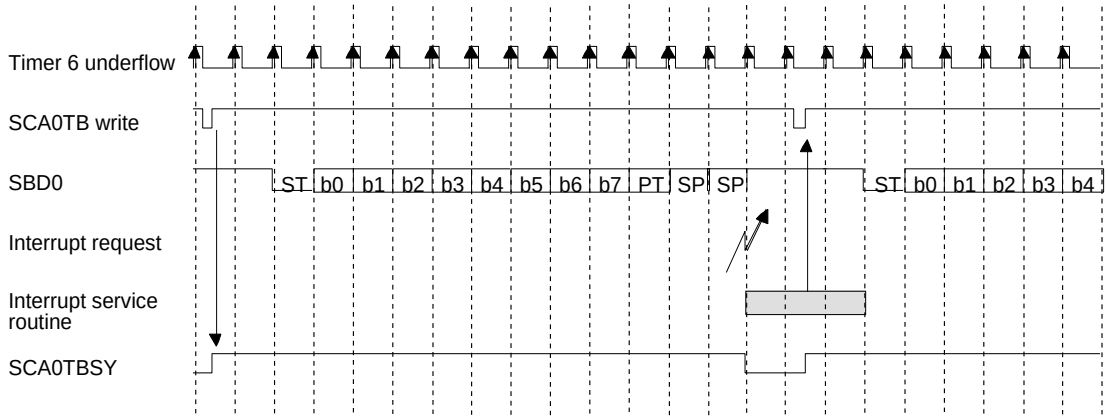


Figure 6-3-11 Asynchronous Bit Transmission Timing

■ Serial Reception in Synchronous Mode

This section describes the serial interface 0 reception in synchronous mode with LSB first bit order, 8-bit data transfer and odd parity settings.

■ Serial Interface 0 Setting

(1) Set the operating conditions in the serial 0 control register (SCA0CTR). Select synchronous mode, LSB first bit order, 8-bit data transfer and odd parity.

SCA0CTR x'E00200'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
0	0	0	0	1	0	0	0	1	1	1	1	0	-	0	0

(2) Set the transmit/receive enable bits (receive enable bits or transmit/receive enable bits) of the serial control register (SCA0CTR).

SCA0CTR x'E00200'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
0	1	0	0	1	0	0	0	1	1	1	1	0	-	0	0

6-3-3 I²C Transmission (Serial 0 and Serial 1)

This section describes the I²C transmission using the serial interface 0.
Master transmission is operated using SDA0 and SCL0 pins.

■ Initial Setting

(1) Set the SDA0 and SCL0 pins to open drain with the port A mode control registers (PAHMD).

PAHMD: x'E0074D'

7	6	5	4	3	2	1	0
-	-	-	-	P8 LMD3	P8 LMD2	P8 LMD1	P8 LMD0
0	0	0	0	1	1	1	0

(2) Set the serial 0 control register (SCA0TR). Select 8-bit character length, I²C protocol, I²C mode on. The parity bit is set to 1 to output '1' to ACK, and enable both transmission and reception enable flags, disable the break.

SCA0CTR: x'E00200'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
1	1	0	0	1	0	1	1	1	1	0	1	0	-	1	1

ACK is set by the parity bits. To output '1' to ACK, select 1 by the parity bits. To output '0' to ACK, select 0 by the parity bits. To output none to ACK, select none by the parity bits.

■ Start Sequence Transmission

(3) Write '1' to the I²C sequence output flag (SCA0IIC) of the SCA0CTR register. This set the SDA0 pin output to low. When the start sequence occurs correctly, the I²C detection flag (SCA0IST) of the serial 0 status register (SCA0STR1) becomes '1'. The arbitration lost detection cannot be performed even though the start sequence exists at the same time.

Enabling reception detects the start sequence.

■ Data Transmission 1

(4) Write the data to the serial 0 transmission register (SCA0TB). This allows the data to output. The SDA0 pin output changes with 1/16 cycle delay of the falling edge of the SCL0 pin output.

(5) After transmission ends, SDA0 pin output and SCL0 pin output stay low.

■ Data Transmission 2

(6) To transmit the data continuously, Write the data to the serial transmission register.

■ Stop Bit Transmission

(7) Set the PAOUT4 flag of the port A output register (PAOUT) to '0'.

(8) Set the PADIR4 flag of the port A input/output control register (PADIR) to '1'.

(9) Set the PAHMD1 and PAHMD0 flags of the port A mode register H (PAHMD) to '00'. Switch the PA4 pin to the port function.

(10) Write '0' to the SCA0IIC flag of the SCA0CTR register to end the data transmission. Do not write during transmission.

(11) Set the SCL0 pin output to high as soon as the SCA0IIC flag is written. The SCA0ISP flag of the SCA0STR1 register becomes '1'. The SCA0IST and SCA0ISP flags of the SCA0STR1 register are cleared by writing/reading of the serial transmission/reception buffer.

(12) Wait for 4 μs.

(13) Set the PAHMD1 and AHMD0 flags of the port A mode register H (PAHMD)to '10'. Switch the PA4 pin to the SDA0 open drain input/output function. It transmits the stop sequence.

The following figure is the master transmission timing.

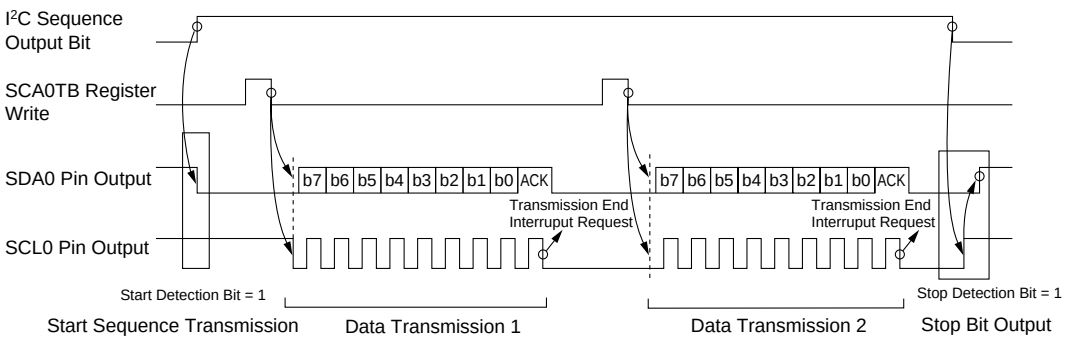

I²C Sequence Output Bit
SCA0TB Register Write
SDA0 Pin Output
SCL0 Pin Output
b7 b6 b5 b4 b3 b2 b1 b0 ACK
b7 b6 b5 b4 b3 b2 b1 b0 ACK
Transmission End Interruput Request
Transmission End Interruput Request
Start Detection Bit = 1
Stop Detection Bit = 1
Start Sequence Transmission
Data Transmission 1
Data Transmission 2
Stop Bit Output

Figure 6-3-12 Master Transmission Timing (with ACK)

6-3-4 I²C Reception (Serial 0 and Serial 1)

This section describes the I²C reception using the serial interface 0.
Master reception is operated using SDA0 and SCL0 pins.

To enter the master reception mode, the first 1 byte must be transmitted during master transmission. Therefore, master reception is performed during the interrupt service routine which runs after the data has been transferred. Refer to “6-3-3 I²C Transmission” for master transmission.

■ Initial Setting

(1) Enable the reception enable flag (SCA0REN) of the serial 0 control register (SCA0CTR) during the serial transmission end interrupt service routine.

SAC0CTR: x’E00200’

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCA0 TEN	SCA0 REN	SCA0 BRE	SCA0 IIC	SCA0 PTL	SCA0 TI	SCA0 OD	SCA0 ICM	SCA0 LN	SCA0 PTY2	SCA0 PTY1	SCA0 PTY0	SCA0 SB	-	SCA0 S1	SCA0 S0
1	1	0	0	1	0	1	1	1	1	0	1	0	-	1	1

The step is not required when reception is enabled by the initial setting.

This step can be omitted if it is the same setting in transmission.

ACK is set by the parity bits. To output ‘1’ to ACK, select ‘1’ by the parity bits. To output ‘0’ to ACK, select ‘0’ by the parity bits. To output none to ACK, select none by the parity bits.

■ Data Reception

(2) Write x’FF’ as the dummy data to the serial 0 transmission buffer register (SCA0TB). Herewith the SDA0 pin output becomes ‘H’ and the master reception is operated.

(3) In the serial reception interrupt routine, read the data from the serial reception register. (The transmission interrupt and the reception interrupt are interchangeable.)

(4) To receive the data continuously, write x’FF’ as the dummy data to the SCA0TB register.

■ Stop Sequence

(5) Set the PAOUT4 flag of the port A output register (PAOUT) to '0'.

(6) Set the PADIR4 flag of the port A input/output control register (PADIR) to '1'.

(7) Set the PAHMD1 and PAHMD0 flags of the port A mode register H (PAHMD) to '00'. Switch the PA4 pin to the port function.

(8) Write '0' to the SCA0IIC flag of the SCA0CTR register.

(9) Set the SCL0 pin output to high as soon as the SCA0IIC flag is written. The SCA0ISP flag of the SCA0STR1 register becomes '1'. The SCA0IST and SCA0ISP flags of the SCA0STR1 register are cleared by writing/reading of the serial transmission/reception buffer.

(10) Wait for 4 μs.

(11) Set the PAHMD1 and PAHMD0 flags of the port A mode register H to '10'. Switch the PA4 pin to the SDA0 open drain input/output function. It outputs the stop sequence. After output of the stop sequence, initialize the reception by disabling the reception enable.

The following figure is the master reception timing.

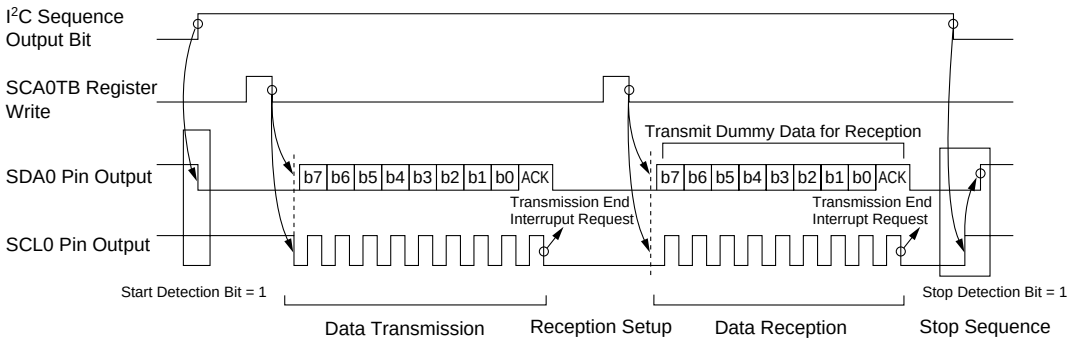

I2C Sequence Output Bit
SCA0TB Register Write
SDA0 Pin Output
SCL0 Pin Output
Transmit Dummy Data for Reception
b7 b6 b5 b4 b3 b2 b1 b0 ACK
b7 b6 b5 b4 b3 b2 b1 b0 ACK
Transmission End Interruput Request
Transmission End Interrupt Request
Start Detection Bit = 1
Stop Detection Bit = 1
Data Transmission
Reception Setup
Data Reception
Stop Sequence

Figure 6-3-13 Master Reception Timing (with ACK)

Chapter 7 Analog Interface

7-1 Analog Interface

The MN102H74G/74F/74D/F74G contains a 10-bit charge redistribution A/D converter.

7-1-1 Analog Interface Function

The following table shows functions of A/D converter.

Table 7-1-1 A/D Converter Functions

A/D Input Pin	Eight pins
Pin Name	AN7 to AN0
Resolution	10 bit The A/D converter divides the voltage between V_{DD} and V_{SS} into 1024, and stores this converted result in ANnBUF.
Conversion Time	2 μs (with 12 MHz oscillator)
Operating Mode	30 operating modes: Single conversion of channel n (n = 0 to 7) Single conversion of channel 0 to channel m (m = 1 to 7) Continuous conversion of channel n (n = 0 to 7) Continuous conversion of channel 0 to channel m (m = 1 to 7)
Conversion Start	Timer 2 underflow or register setting
Interrupt	An interrupt occurs each time the conversion sequence ends
Low-power Mode Function	Internal ladder resistor on/off

The AN pin is corresponded to channel number. For example, the AN3 pin is corresponded to channel 3.

7-1-2 Analog Interface Configuration

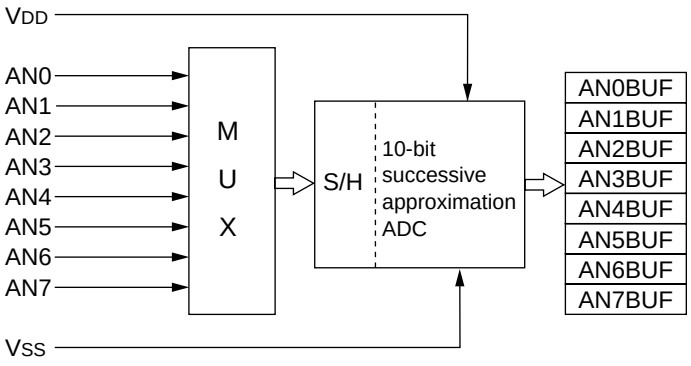


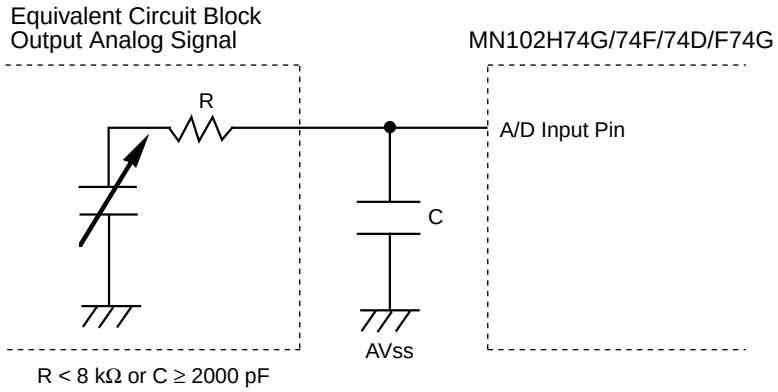
Figure 7-1-1 Analog Interface Configuration

■ Cautions on Operation of A/D Converter

(1) Set the analog signal impedance converted A/D to less or equal 8 kΩ.

(2) Otherwise, to suppress the voltage variation of A/D input pin, connect 2000 pF or more capacitor to A/D input pin.

(3) During A/D conversion, to prevent the voltage variation of A/D input pin, do not change the output level of the microcontroller H to L or L to H, or do not switch ON/OFF the peripheral load circuit.



7-2 Control Registers

The A/D converter contains the A/D conversion control register (ANCTR) and the A/D conversion data buffers (ANnBUF) corresponded to channel 7 to channel 0 (AN7 pin to AN0 pin).

7-2-1 List of Analog Interface Control Registers

The following table shows registers to control analog interface.

Table 7-2-1 List of Analog Interface Control Registers

Register	Address	R/W	Function	Reference
ANCTR	x'E00300'	R/W	A/D Conversion Control Regitser	7-5
AN0BUF	x'E00310'	R	A/D0 Conversion Data Buffer	7-6
AN1BUF	x'E00312'	R	A/D1 Conversion Data Buffer	7-6
AN2BUF	x'E00314'	R	A/D2 Conversion Data Buffer	7-6
AN3BUF	x'E00316'	R	A/D3 Conversion Data Buffer	7-7
AN4BUF	x'E00318'	R	A/D4 Conversion Data Buffer	7-7
AN5BUF	x'E0031A'	R	A/D5 Conversion Data Buffer	7-7
AN6BUF	x'E0031C'	R	A/D6 Conversion Data Buffer	7-8
AN7BUF	x'E0031E'	R	A/D7 Conversion Data Buffer	7-8

Note R/W: Read/Write
R : Read only

The A/D conversion control register (ANCTR) sets the A/D conversion operating conditions.
The A/D conversion data buffers (ANnBUF) store the A/D conversion results of channel 7 to channel 0 (AN7 pin to AN0 pin). Therefore, these registers cannot be written.

7-2-2 A/D Conversion Control Register

■ A/D Conversion Control Register (ANCTR)

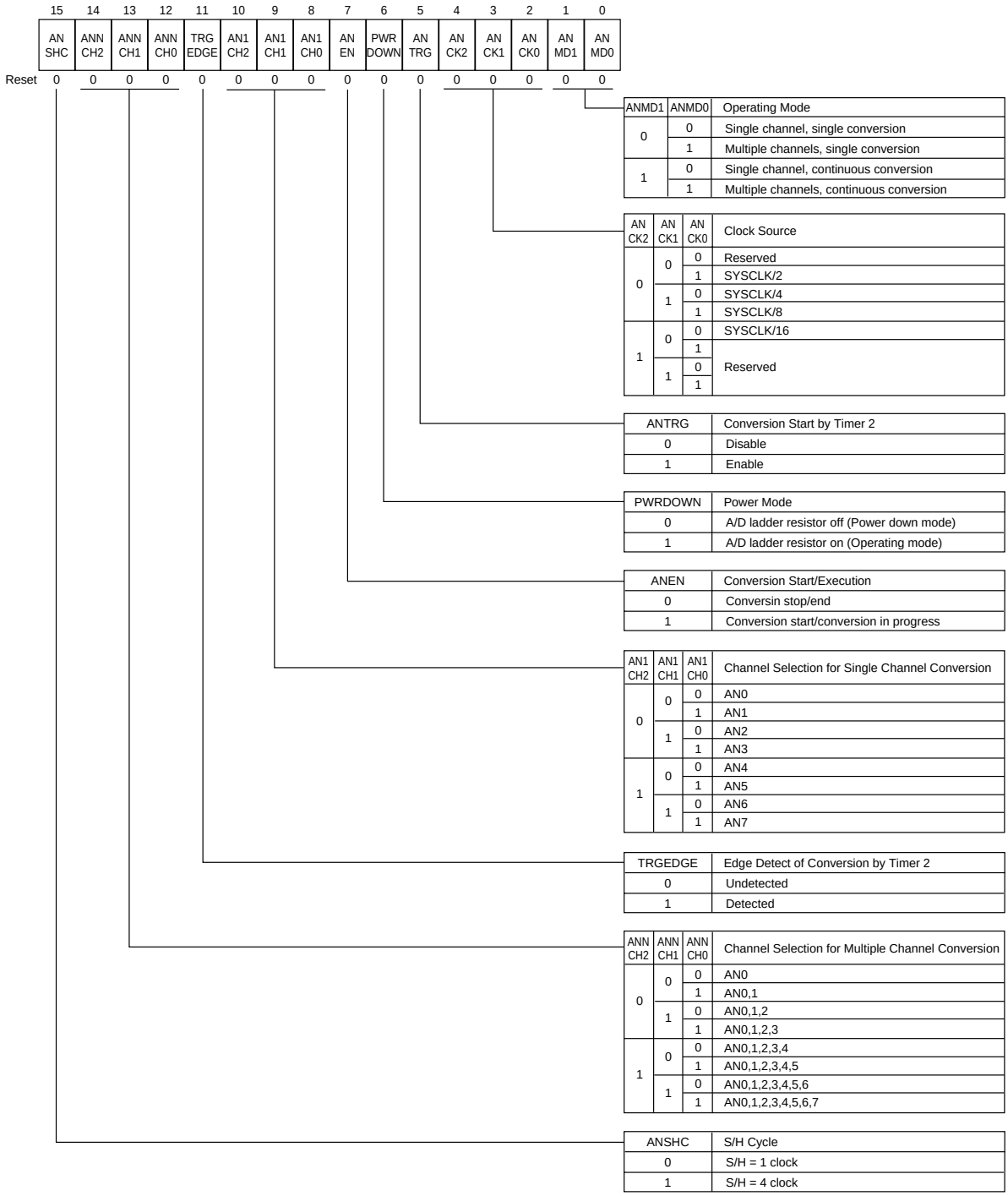


Figure 7-2-1 A/D Conversion Control Register (ANCTR: x'E00300', R/W)

7-2-3 A/D Conversion Data Buffer

■ A/D0 Conversion Data Buffer (AN0BUF)

The AN0BUF is a read-only register. AN0BUF[9:0] : The A/D conversion result of channel 0 (AN0 pin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AN0BUF	—	—	—	—	—	—	AN0 BUF9	AN0 BUF8	AN0 BUF7	AN0 BUF6	AN0 BUF5	AN0 BUF4	AN0 BUF3	AN0 BUF2	AN0 BUF1	AN0 BUF0	(Reset : 00000000 00000000)

Figure 7-2-2 A/D 0 Conversion Data Buffer (AN0BUF: xE00310, R)

■ A/D1 Conversion Data Buffer (AN1BUF)

The AN1BUF is a read-only register. AN1BUF[9:0] : The A/D conversion result of channel 1 (AN1 pin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AN1BUF	—	—	—	—	—	—	AN1 BUF9	AN1 BUF8	AN1 BUF7	AN1 BUF6	AN1 BUF5	AN1 BUF4	AN1 BUF3	AN1 BUF2	AN1 BUF1	AN1 BUF0	(Reset : 00000000 00000000)

Figure 7-2-3 A/D1 Conversion Data Buffer (AN1BUF: xE00312, R)

■ A/D2 Conversion Data Buffer (AN2BUF)

The AN2BUF is a read-only register. AN2BUF[9:0] : The A/D conversion result of channel 2 (AN2 pin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AN2BUF	—	—	—	—	—	—	AN2 BUF9	AN2 BUF8	AN2 BUF7	AN2 BUF6	AN2 BUF5	AN2 BUF4	AN2 BUF3	AN2 BUF2	AN2 BUF1	AN2 BUF0	(Reset : 00000000 00000000)

Figure 7-2-4 A/D2 Conversion Data Buffer (AN2BUF: xE00314, R)

■ A/D3 Conversion Data Buffer (AN3BUF)

The AN3BUF is a read-only register. AN3BUF[9:0] : The A/D conversion result of channel 3 (AN3 pin)

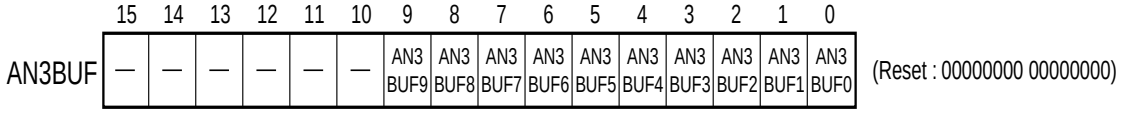

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AN3BUF
— — — — — — AN3 BUF9 AN3 BUF8 AN3 BUF7 AN3 BUF6 AN3 BUF5 AN3 BUF4 AN3 BUF3 AN3 BUF2 AN3 BUF1 AN3 BUF0
(Reset : 00000000 00000000)

Figure 7-2-5 A/D3 Conversion Data Buffer (AN3BUF: xE00316, R)

■ A/D4 Conversion Data Buffer (AN4BUF)

The AN4BUF is a read-only register. AN4BUF[9:0] : The A/D conversion result of channel 4 (AN4 pin)

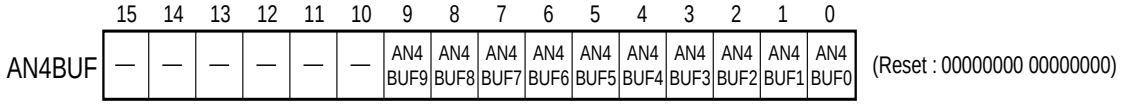

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AN4BUF
— — — — — — AN4 BUF9 AN4 BUF8 AN4 BUF7 AN4 BUF6 AN4 BUF5 AN4 BUF4 AN4 BUF3 AN4 BUF2 AN4 BUF1 AN4 BUF0
(Reset : 00000000 00000000)

Figure 7-2-6 A/D4 Conversion Data Buffer (AN4BUF: xE00318, R)

■ A/D5 Conversion Data Buffer (AN5BUF)

The AN5BUF is a read-only register. AN5BUF[9:0] : The A/D conversion result of channel 5 (AN5 pin)

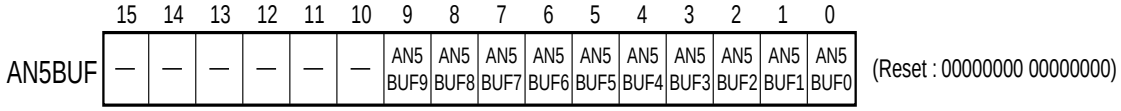

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AN5BUF
— — — — — — AN5 BUF9 AN5 BUF8 AN5 BUF7 AN5 BUF6 AN5 BUF5 AN5 BUF4 AN5 BUF3 AN5 BUF2 AN5 BUF1 AN5 BUF0
(Reset : 00000000 00000000)

Figure 7-2-7 A/D5 Conversion Data Buffer (AN5BUF: xE0031A, R)

■ A/D6 Conversion Data Buffer (AN6BUF)

The AN6BUF is a read-only register. AN6BUF[9:0] : The A/D conversion result of channel 6 (AN6 pin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AN6BUF	—	—	—	—	—	—	AN6 BUF9	AN6 BUF8	AN6 BUF7	AN6 BUF6	AN6 BUF5	AN6 BUF4	AN6 BUF3	AN6 BUF2	AN6 BUF1	AN6 BUF0	(Reset : 00000000 00000000)

Figure 7-2-8 A/D6 Conversion Data Buffer (AN6BUF: xE0031C, R)

■ A/D7 Conversion Data Buffer (AN7BUF)

The AN7BUF is a read-only register. AN7BUF[9:0] : The A/D conversion result of channel 7 (AN7 pin)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AN7BUF	—	—	—	—	—	—	AN7 BUF9	AN7 BUF8	AN7 BUF7	AN7 BUF6	AN7 BUF5	AN7 BUF4	AN7 BUF3	AN7 BUF2	AN7 BUF1	AN7 BUF0	(Reset : 00000000 00000000)

Figure 7-2-9 A/D7 Conversion Data Buffer (AN7BUF: xE0031E, R)

7-3 A/D Converter Operation

7-3-1 A/D Converter Operation

■ A/D Converter Clock Source Setup

The A/D converter clock source is selected from SYSCLK/2, SYSCLK/4, SYSCLK/8 or SYSCLK/16. The conversion time is 12 cycles of the A/D converter clock source when S/H = 1 cycle and 10-bit resolution is selected. For example, the conversion time when SYSCLK/4 is selected is [SYSCLK cycle × 4 (divisor) × 12 (cycles)].

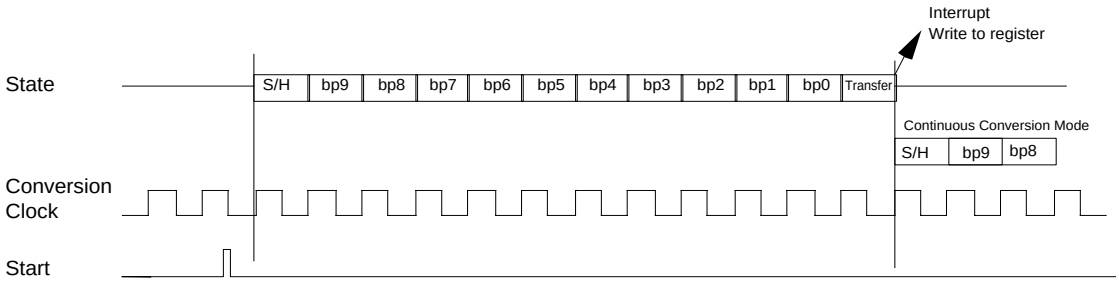


Figure 7-3-1 A/D Conversion Timing (S/H = 1 cycle, 10-bit resolution)

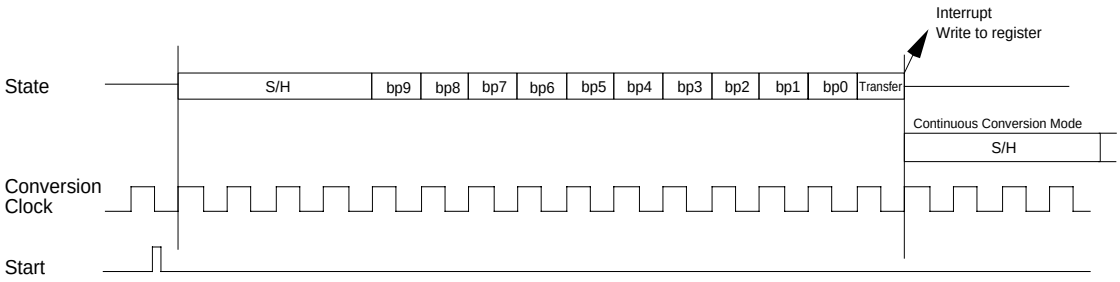


Figure 7-3-2 A/D Conversion Timing (S/H = 4 cycle, 10-bit resolution)

■ Single Channel/Single Conversion

When the ANMD[1:0] flags of the ANCTR register are set to 00 (single channel, single conversion), the AD converter converts one AN input signal once. An interrupt occurs when the conversion ends. The number of channel to be converted is set to the AN1CH[2:0] flags. When the A/D converter starts using the ANEN flag, set the ANTRG flag to 0 and the ANEN flag to 1. The ANEN flag becomes 0 after the conversion ends.

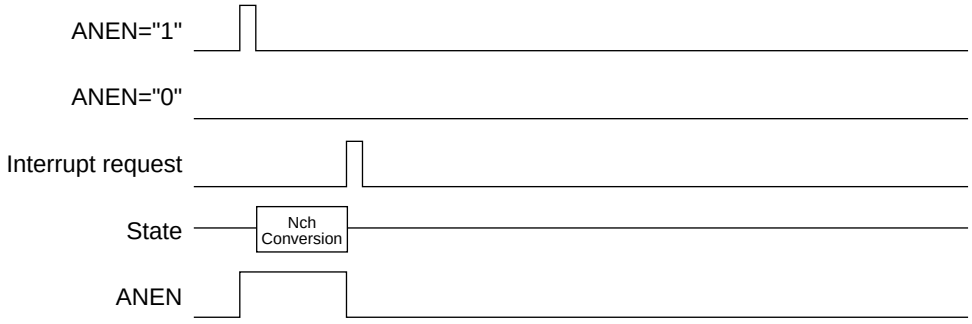


Figure 7-3-3 Single Channel/Single Conversion Timing

■ Multiple Channels/Single Conversion

When the ANMD[1:0] flags of the ANCTR register are set to 01 (multiple channels, single conversion), the AD converter converts consecutive AN input signals once. An interrupt occurs when the conversion sequence ends. The AN1CH[2:0] flags are set to channel 0 and the ANNCH[2:0] flags are set to the last channel to be converted. The conversion always starts with channel 0. When the A/D converter starts using the ANEN flag, set the ANTRG flag to 0 and the ANEN flag to 1. The ANEN flag becomes 0 after the conversion sequence ends. The AN1CH[2:0] flags show the number of channel being converted and cleared to 0 when the conversion sequence ends.

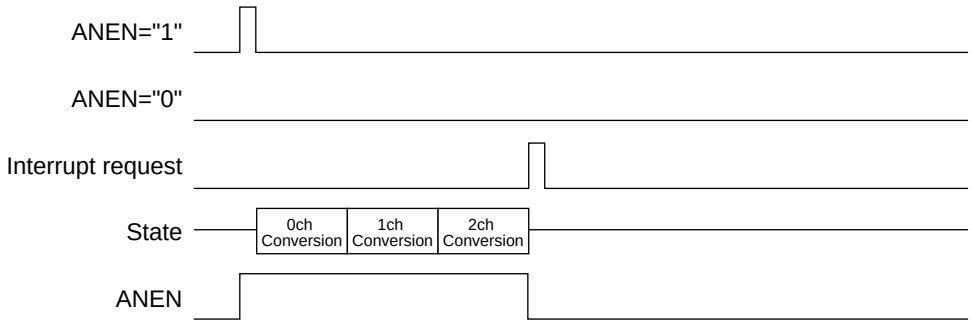


Figure 7-3-4 Multiple Channels/Single Conversion Timing

■ Single Channel/Continuous Conversion

When the ANMD[1:0] flags of the ANCTR register are set to 10 (single channel, continuous conversion), the AD converter converts one AN input signal continuously. An interrupt occurs when the conversion ends. The number of channel to be converted is set to the AN1CH[2:0] flags. When the A/D converter starts using the ANEN flag, set the ANTRG flag to 0 and the ANEN flag to 1. Setting the ANEN flag to 0 forcible by program ends the conversion. The A/D converter halts the operation before completing the operation if the conversion is in progress.

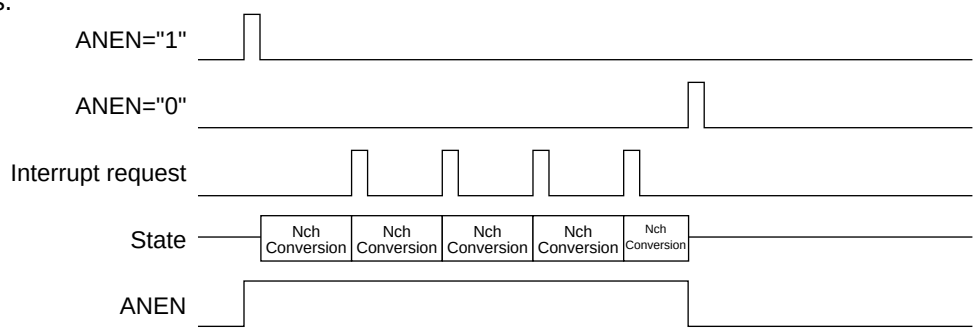


Figure 7-3-5 Single Channel/Continuous Conversion Timing

Ignore setting of ANNCH[2:0] flags.

AN1CHn, ANNCHn, ANEN and ANTRG are flags of the A/D conversion control register.

■ Multiple Channels/Continuous Conversion

When the ANMD[1:0] flags of the ANCTR register are set to 11 (multiple channels, continuous conversion), the AD converter converts multiple, consecutive AN input signals continuously. An interrupt occurs when the conversion sequence ends. The AN1CH[2:0] flags are set to channel 0 and the ANNCH[2:0] flags are set to the last channel to be converted. The conversion always starts with channel 0. When the A/D converter starts using the ANEN flag, set the ANTRG flag to 0 and the ANEN flag to 1. Setting the ANEN flag to 0 forcible by program ends the conversion. The A/D converter halts the operation before completing the operation if the conversion is in progress. The AN1CH[2:0] flags show the number of channel being converted and cleared to 0 when the conversion sequence ends.

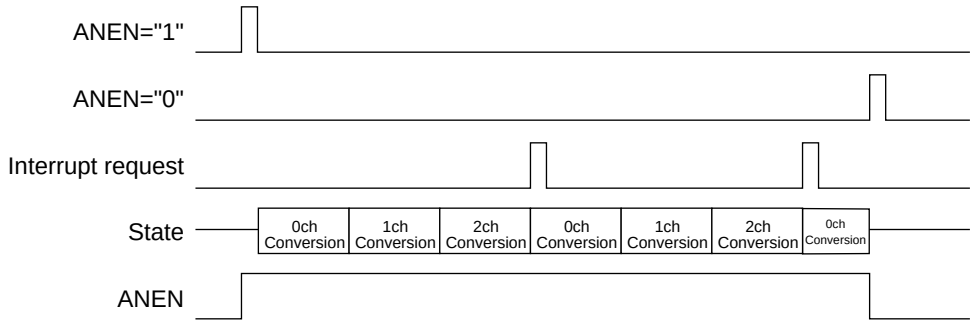


Figure 7-3-6 Multiple Channels/Continuous Conversion Timing

■ A/D Conversion Input Pin Setup

The AN1CH[2:0] flags of the ANCTR register select the input pin for A/D conversion.

Table 7-3-1 A/D Conversion Input Pin Setup

AN1CH2	AN1CH1	AN1CH0	A/D Pin
0	0	0	AN0
		1	AN1
	1	0	AN2
		1	AN3
1	0	0	AN4
		1	AN5
	1	0	AN6
		1	AN7

■ A/D Conversion Clock Setup

The ANCK[2:0] flags of the ANCTR register select the clock for A/D conversion.

Table 7-3-2 A/D Conversion Clock Setup

ANCK2	ANCK1	ANCK0	A/D Conversion Clock
0	0	0	Reserved
		1	SYSCLK/2
	1	0	SYSCLK/4
		1	SYSCLK/8
1	0	0	SYSCLK/16
		1	Reserved
	1	0	
		1	

■ A/D Conversion Sampling Cycle Setup

The ANSHC flag of the ANCTR register sets the sampling time for A/D conversion. The appropriate value must be set based on the analog input impedance because the A/D conversion sampling time changes depending on the external circuit.

Table 7-3-3 A/D Conversion Sampling Time Setup

ANSHC	Sampling/Hold Time (Cycle)
0	ADCK × 1
1	ADCK × 4

■ A/D Ladder Resistor Setup

The current is flowed to the ladder resistor and makes the A/D conversion standby by setting the PWRDOWN flag of the ANCTR register to 1. When the A/D converter stops, setting the PWRDOWN flag of the ANCTR register to 0 reduces the power consumption.

Table 7-3-4 A/D Ladder Resistor Setup

PWRDOWN	A/D Ladder Resistor Control
0	A/D ladder resistor off (Power down mode)
1	A/D ladder resistor on (Operating mode)

7-3-2 A/D Operation Setup Example

■ Single Channel A/D Conversion Using AN6 Pin

This section describes the A/D conversion by software. The MN102H74G/74F/74D/F74G inputs the analog voltage (0 V to 3.3 V) from the AN6 pin and obtains the A/D conversion results.

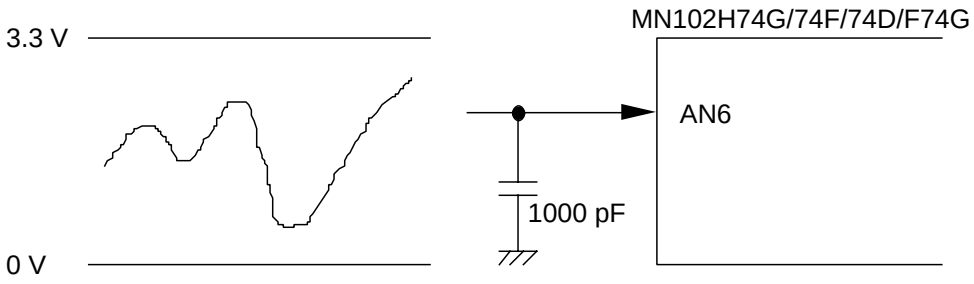


Figure 7-3-7 Single Channel A/D Conversion

The following table lists the related A/D registers.

Table 7-3-5 List of A/D Conversion Control Registers for Setup

Register	Address	R/W	Function	Reference
ANCTR	x'E00300'	R/W	A/D Conversion Control Register	7-5
AN6BUF	x'E0031C'	R	A/D6 Conversion Data Buffer	7-8

(1) Set the A/D operating conditions to the A/D conversion control register (ANCTR). Select single channel/ single conversion. Select SYSCLK/4 as clock source. Set the conversion start/execution flag (ANEN) to 0 and the AN1CH[2:0] flags to 110 (channel 6). Set the PWRDOWN flag to 1 (operating mode) and ANSHC flag to 0 (1 cycle).

ANCTR: x'E00300'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANSHC	ANNCH2	ANNCH1	ANNCH0	TRGEDGE	AN1CH2	AN1CH1	AN1CH0	ANEN	PWRDOWN	ANTRG	ANCK2	ANCK1	ANCK0	ANMD1	ANMD0
0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0

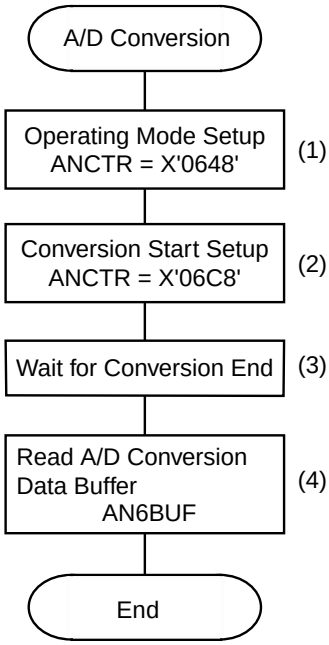
Ignore setting of ANNCH[2:0] flags.

(2) Start the A/D converter by setting the ANEN flag to 1. The conversion starts on the first rise edge of the A/D conversion clock after setting the ANEN flag to 1. The conversion time is 12 cycles of the A/D conversion clock.

Set the ANEN flag to 1 when starting the A/D converter with software.

(3) Wait for the conversion end. The ANEN flag is 1 during the conversion and is cleared to 0 when the conversion ends. The program waits until the ANEN flag is cleared to 0.

The conversion result is read by occurring an interrupt. In this case, an interrupt occurs after the conversion result is stored in the AN6BUF buffer. Therefore, the program does not need to wait for the ANEN clear.



(4) Read the A/D6 conversion data buffer (AN6BUF). The converter divides the voltage of 0 V and 3.3 V into 1024 steps and stores the conversion result of 0 to 1023 to the AN6BUF buffer.

AN6BUF: x'E0031C'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	AN6BUF9	AN6BUF8	AN6BUF7	AN6BUF6	AN6BUF5	AN6BUF4	AN6BUF3	AN6BUF2	AN6BUF1	AN6BUF0

■ Multiple Channel A/D Conversion Using AN2 Pin to AN0 Pin

The analog voltage (0 V to 3.3 V) is input from AN2 pin, AN1 pin and AN0 pin and the A/D conversion results is obtained. The A/D converter operates periodically using timer 2.

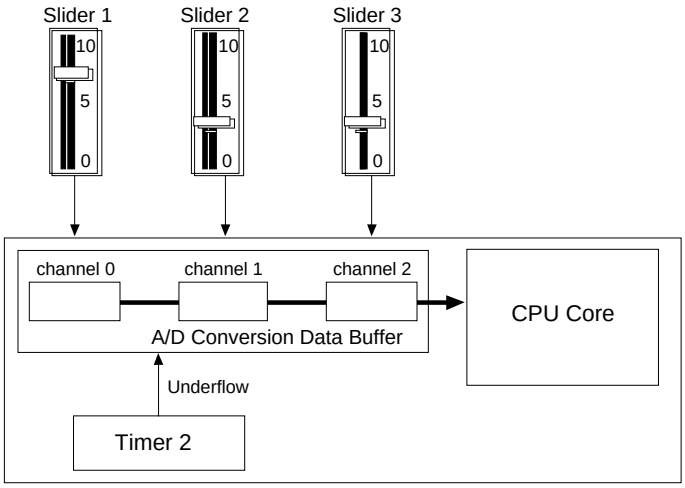


Figure 7-3-8 Multiple Channel A/D Conversion

The following table lists the related A/D registers.

Table 7-3-6 List of A/D Conversion Control Registers for Setup

Register	Address	R/W	Function	Reference
ANCTR	x'E00300'	R/W	A/D Conversion Control Register	7-5
AN0BUF	x'E00310'	R	A/D0 Conversion Data Buffer	7-6
AN1BUF	x'E00312'	R	A/D1 Conversion Data Buffer	7-6
AN2BUF	x'E00314'	R	A/D2 Conversion Data Buffer	7-6

(1) Set the A/D operating conditions to the A/D conversion control register (ANCTR). Select multiple channels/single conversion. Select SYSCLK/4 as clock source. Set the ANEN flag and the ANTRG flag to 0 and 1 respectively. Set the AN1CH[2:0] flags to 000 (channel 0) and the ANNCH[2:0] flags to '010' (channel 2). Set the ANSHC flag to 0 (1 cycle).

ANCTR: x'E00300'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANSHC	ANNCH2	ANNCH1	ANNCH0	TRGEDGE	AN1CH2	AN1CH1	AN1CH0	ANEN	PWRDOWN	ANTRG	ANCK2	ANCK1	ANCK0	ANMD1	ANMD0
0	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1

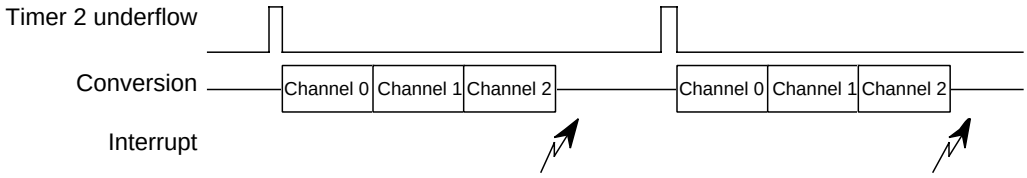


Figure 7-3-9 Multiple Channel A/D Conversion Timing

(2) Wait for conversion end. The conversion ends by generating an interrupt.

(3) Read the AN2BUF buffer to the AN0BUF buffer.

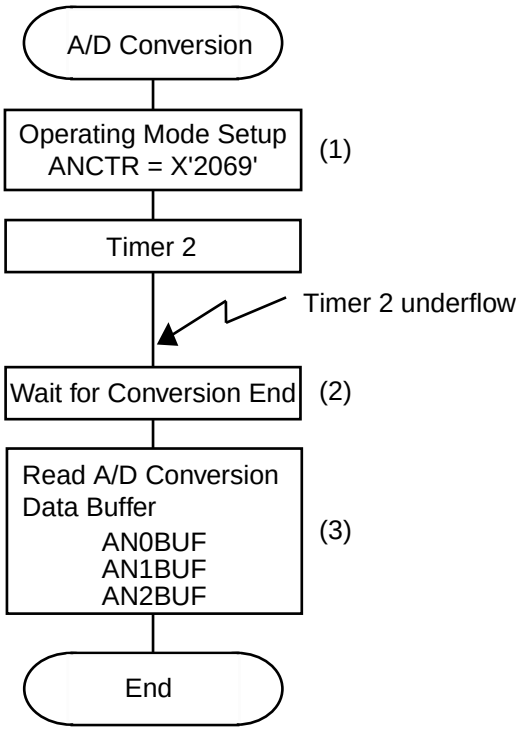


Figure 7-3-10 A/D Conversion Flow

Chapter 8 USB

8-1 Description

8-1-1 USB Function

USB device core with endpoint zero and eight endpoints is a full-speed function core. Each of the eight endpoints is capable of handling bulk, interrupt, and isochronous data transfers. This core provides a high-level user interface, shielding the user from USB protocol details. It supports automatic data retry, data toggle, and all power management functions such as suspend and resume. Table 8-1-1 lists the maximum FIFO sizes for each endpoint.

Table 8-1-1 Endpoint Specifications

Endpoint	FIFO Size	Transfer Mode
Endpoint 0	64 bytes	Control (IN,OUT)
Endpoint 1	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 2	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 3	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 4	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 5	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 6	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 7	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)
Endpoint 8	128 bytes	Bulk, Interrupt,Isochronous (IN/OUT)

8-2 Features

- Conforms to USB 1.1 standard.
- USB protocol control is not required as a result of high-level interface.
- Complete device configuration is achieved.
- Compatible to Open HCI and Intel UHCI standard.
- Supports 12 Mbps.
- Corresponds to suspend and resume signals.
- Endpoints (EP1 to EP8) are IN and OUT programmable.

8-3 Block Diagram

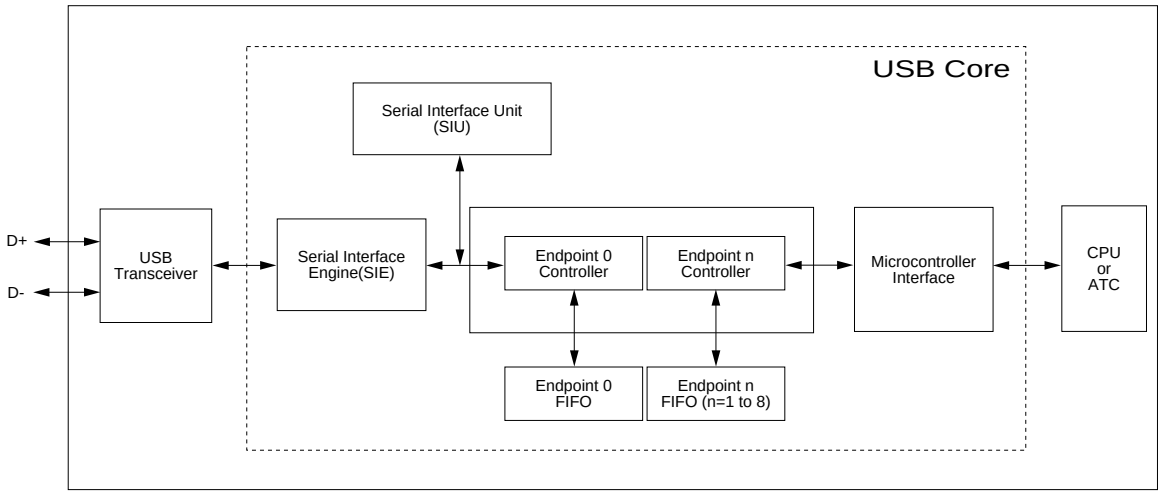


Figure 8-3-1 USB Block Diagram

8-4 USB Special Functions

8-4-1 Double Buffering Function

Setting the maximum packet size (MAXP) register to less than or equal to half the capacity of the FIFO automatically enables the double buffering function. With double buffering, two packets of data can be stored in the FIFO. If this function is used with an IN endpoint, the controller or ATC can write a second data packet to the FIFO while the USB core is transmitting the first packet. With an OUT endpoint, the controller or ATC can read the first packet from the FIFO while the USB core is receiving the second packet. When the MAXP register is set to the half size of FIFO or more, the SINGLE flag of ICSR2 register is valid to use single buffer.

The following figures show the IN transfer of three packets using double buffer function.

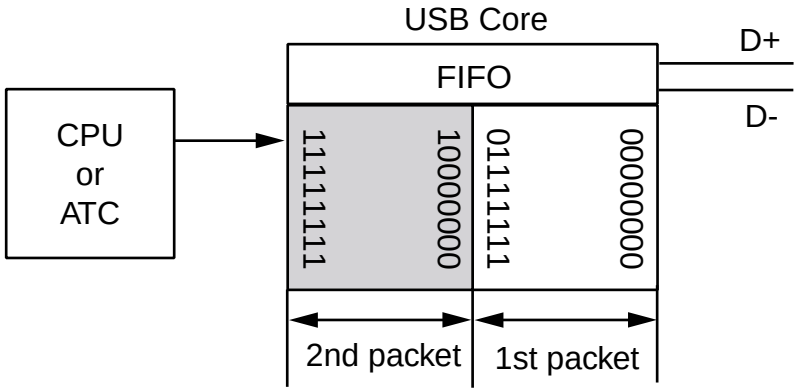


Figure 8-4-1 Second Packet Write during First Packet Transfer

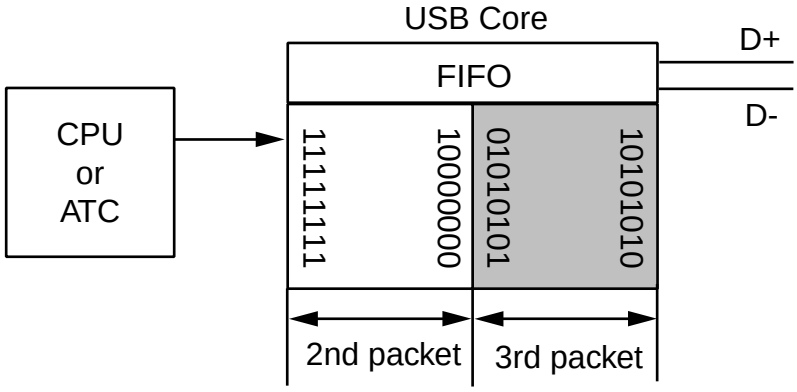


Figure 8-4-2 Third Packet Write during Second Packet Transfer

8-4-2 Data Rate Feedback Function

The data rate feedback function is incorporated to achieve the specification described on page 168 of the USB Specification Rev 1.1. When used in [the data rate feedback] mode, the data toggle bits should be changed after each data packet is sent to the host without regard to the presence or type of handshake packet.

Setting INT_FBACK of the ICSR2 register to 1 selects the data rate feedback mode. When this bit is set to 1, the bulk/interrupt transfer mode is needed to select by setting the ISO bit to zero. The endpoint with the data rate feedback mode does not generate an interrupt when the packet transmission is completed or FIFO_FLUSH is set. When the new data rate to be fed back occurs, write the data in FIFO and IN_PKT_RDY of the ICSR1 register to 1. The USB core sends the same data to IN token until the new data is written to FIFO. The USB core does not clear the packet. The USB core toggles DATA0 and DATA1 even though the USB core does not receive an ACK handshake.

In the data rate feedback mode, when the packet is written in FIFO and is set to MCU_IN_PKT_RDY, the USB core operates FIFO flash once and then clears the previous packet. At the same time, the USB core clears USB_IN_PKT_RDY. After 1 clock cycle the USB core, then clears MCU_PKY_RDY and sets USB_IN_PKT_RDT.

When using the data rate feedback function, verify that the SINGLE flag of the ICSR2 register is 0. Do not use the data rate feedback function in the single buffer mode.

Example interrupt transfer (data rate feedback mode)

MCU: write Data Rate to EP1 FIFO

Write ICSR1=01 (MCU_IN_PKT_RDY set)

Host: **IN <ADDR:00 EP:1>CRC5 <08>**

USB core: **DATA0 ,12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 >CRC16<2f6d>(16Bytes)**

Host: **ACK**

No IRQ received

Host: **IN<ADDR:00 EP:1>CRC5<08>**

USB core: **DATA0 ,12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 >CRC16<2f6d>(16Bytes)**

<No Handshake from Host>

No IRQ received

Host: **IN<ADDR:00 EP:1>CRC5<08>**

USB core: **DATA0 ,12 01 01 00 01 00 00 10 <--Toggle DATA PID!**

d6 04 00 00 00 00 00 00 > CRC16 <2f6d> (16 Bytes)

Host: **ACK**

No IRQ received

Host: **IN<ADDR:00 EP:1>CRC5<08>**

USB core: **DATA1 ,12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 >CRC16<2f6d>(16Bytes)**

Host: **ACK**

No IRQ received

MCU: Write New Data Rate to EP1 FIFO

Write ICSR1=01 (MCU_IN_PKT_RDY set)

Host: **IN<ADDR:00 EP:1>CRC5<08>**

USB core: **DATA0 <18 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 >CRC16<5ael>(16Bytes)**

<No Handshake from Host>

No IRQ received

Host: **IN<ADDR:00 EP:1>CRC5<08>**

USB core: **DATA0 <18 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 >CRC16<5ael>(16Bytes)**

Host: **ACK**

No IRQ received

8-5 ATC Interface

When the endpoint interrupt is used by setting the ATCSW register, interrupt response is accelerated and ATC transferred. Figure 8-5-1 shows connection to the ATC controller. ATACKn and EPIRQn signal is used in each endpoint.

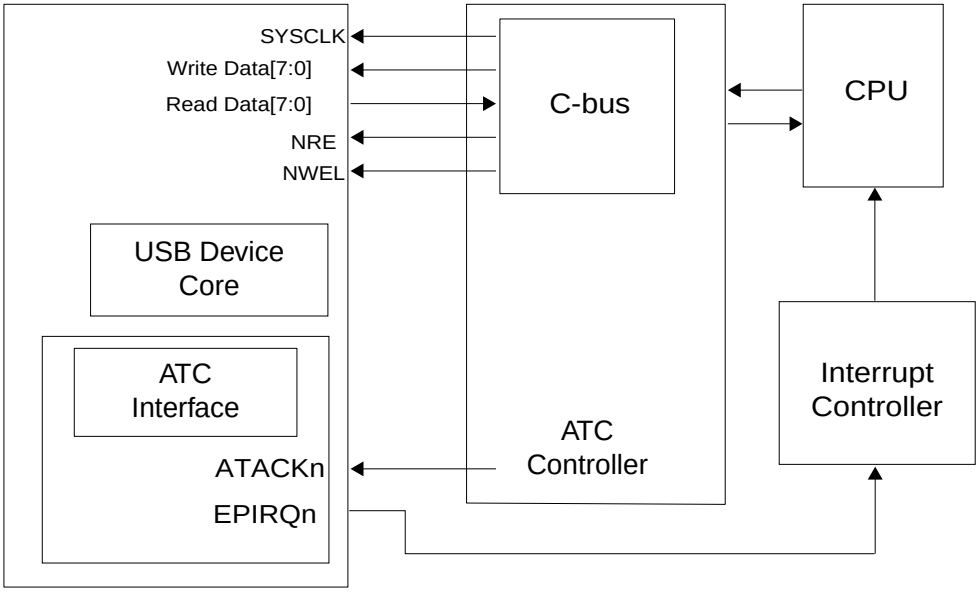


Figure 8-5-1 ATC Interface

When ATC is transferred to the USB core, use burst transfer. One word transfers can not be used. Also set to output ATACKn by ATnCTR and ATnSEL register associated with each endpoint. See “9-2 ATC Control Register”.

> ATC transfer can not be used by hardware activation. In the EPIRQn interrupt service setup, make ATC the software activation.

> In endpoint set in ATC mode by ATCSW register, set zero to AUTO_SET and AUTO_CLR. It is synchronized with falling ATACKn signal, and IN_PKT_RDY is set or OUT_PKT_RDY is cleared automatically. In the endpoint that is not ATC mode, it doesn’t respond the falling of ATACKn.

The condition of EPIRQn output is different from the IN endpoints and the OUT endpoints.

■ In endpoints

When packet transmission ends and IN_PKT_RDY is cleared, EPIRQn is output to associated endpoints.

Note: If there is no packet in the FIFO, EPIRQn is not generated even though the IN token is received. At that time, the USB core returns NAK in bulk-endpoints and does not interrupt.

■ OUT endpoints

When packet is received and OUT_PKT_RDY is set, EPIRQn is output to associated endpoints.

Synchronized with rising edge of ATACKn, IN_PKT_RDY is set in the IN endpoints, and OUT_PKT_RDY is cleared in the OUT endpoints. When ATC function is used, read/write a packet by only one ATC transfer. If a packet is read/written by two or more ATC transfers, the USB core recognizes that two packets are processed.

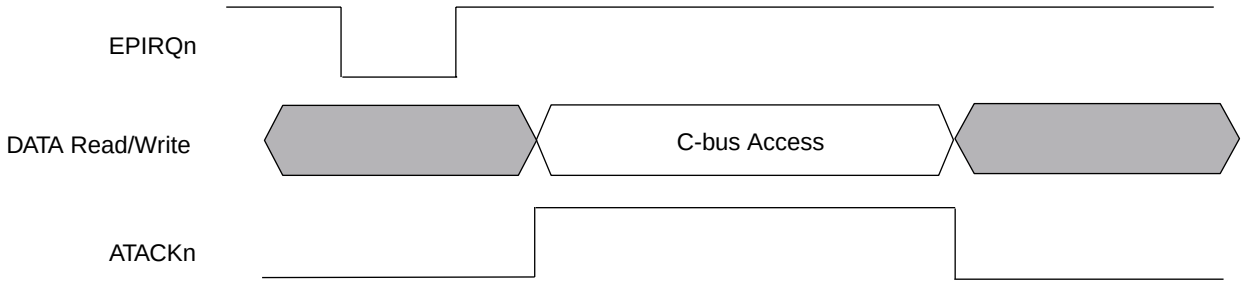


Figure 8-5-2 ATC Transfer

8-5-1 ATC Transfer Mode

ATC transfer has dual and single address access modes. In single access mode, transfer USB ends one cycle of C-BUS access. Figure 8-5-3 through 8-5-6 show mode access timing to IN and OUT endpoints.

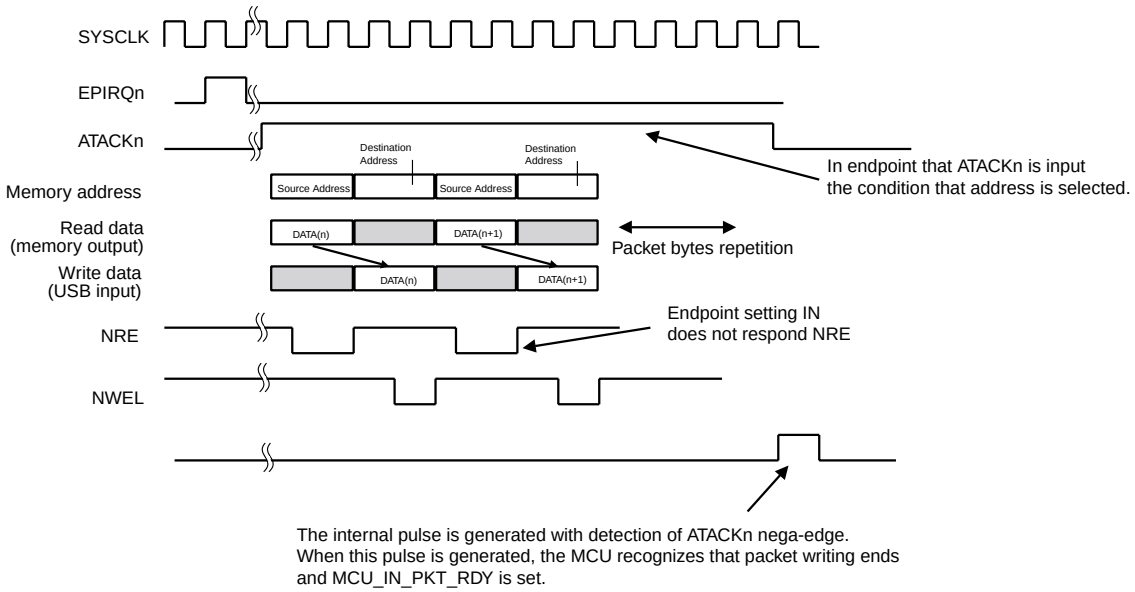


Figure 8-5-3 ATC Transfer Operation for IN Endpoints (Dual Address Mode)

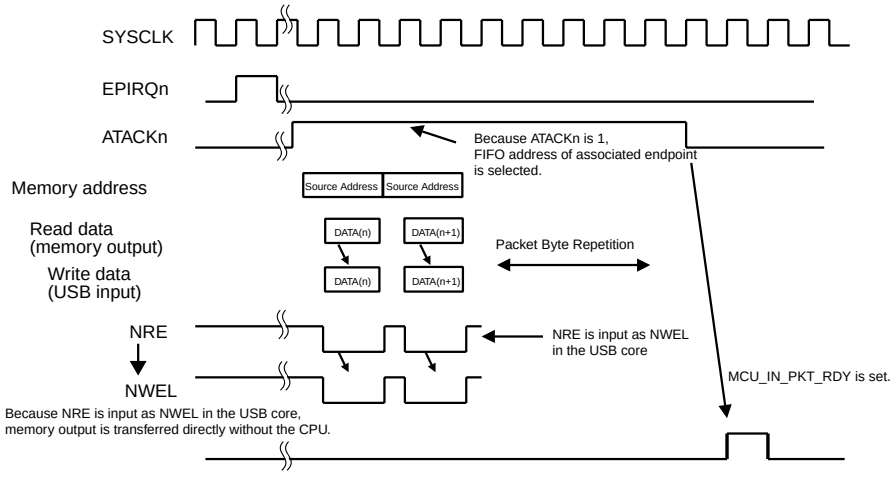


Figure 8-5-4 ATC Transfer Operation for IN Endpoints (Single Address Mode)

Single address mode is enable only in transfer between USB-FIF0 and the internal RAM.

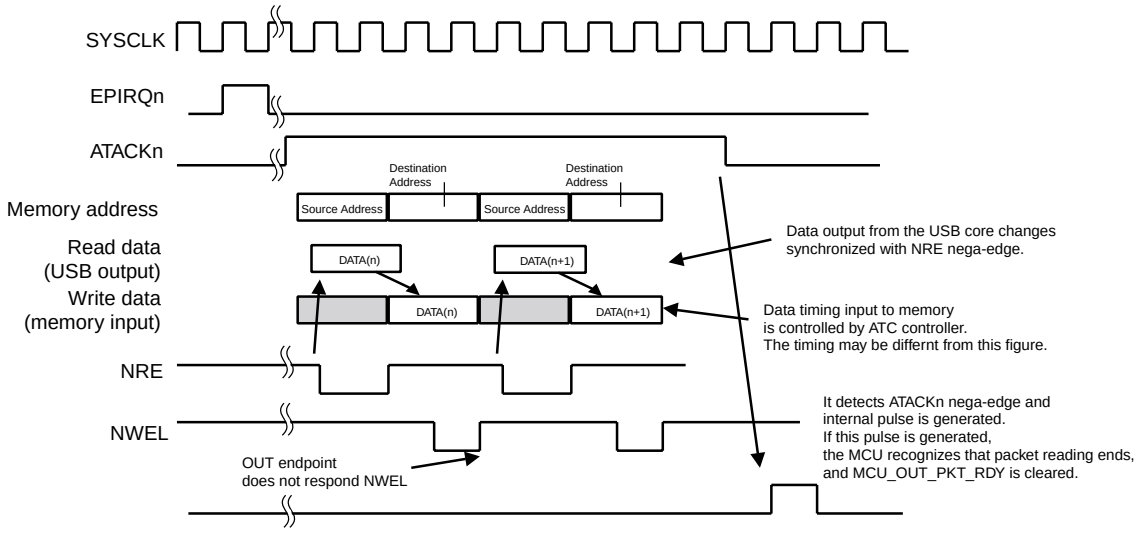

SYSCLK
EPIRQn
ATACKn
Destination Address
Destination Address
Memory address
Source Address
Source Address
Read data (USB output)
DATA(n)
DATA(n+1)
Write data (memory input)
DATA(n)
DATA(n+1)
Data output from the USB core changes synchronized with NRE nega-edge.
Data timing input to memory is controlled by ATC controller. The timing may be differnt from this figure.
NRE
NWEL
It detects ATACKn nega-edge and internal pulse is generated. If this pulse is generated, the MCU recognizes that packet reading ends, and MCU_OUT_PKT_RDY is cleared.
OUT endpoint does not respond NWEL

Figure 8-5-5 ATC Transfer Operation for OUT Endpoints (Dual Address Mode)

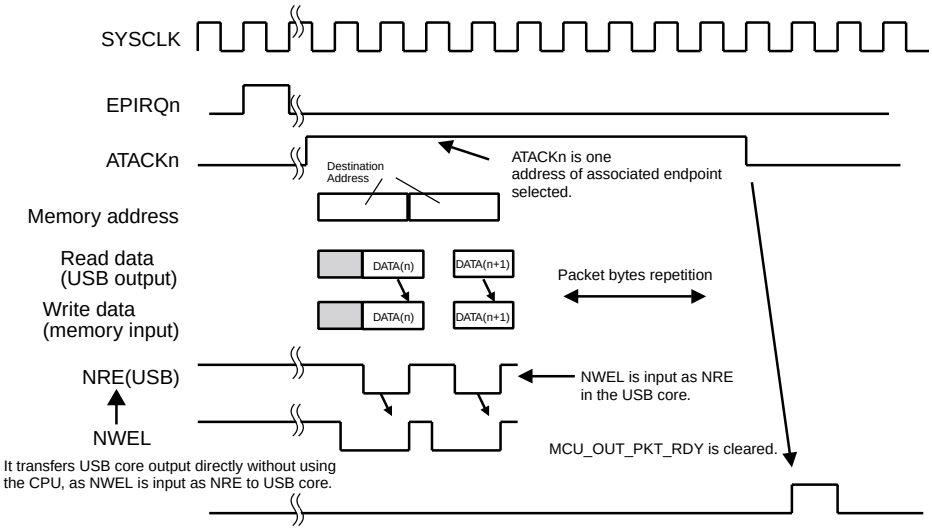


Figure 8-5-6 ATC Transfer Operation for OUT Endpoints (Single Address Mode)

8-5-2 Packet Transfer Using Arbitration Timing Control

This section describes the sequence of using arbitration timing control (ATC). The data packet in USB is variable up to MAX packet size. When ATC is used, transferred packet size must be settled in advance. Set timing as follows.

■ IN transfer

When IN is transferred, if it exceeds packet size of endpoints, it is generally transferred in lots. At that time, the packets may be under the size decided by MAXP of endpoints. When ATC transfer is used, the CPU must examine next packet size at every ATC transfer, and set the number of burst byte when ATC transfer to ATnCNT register.

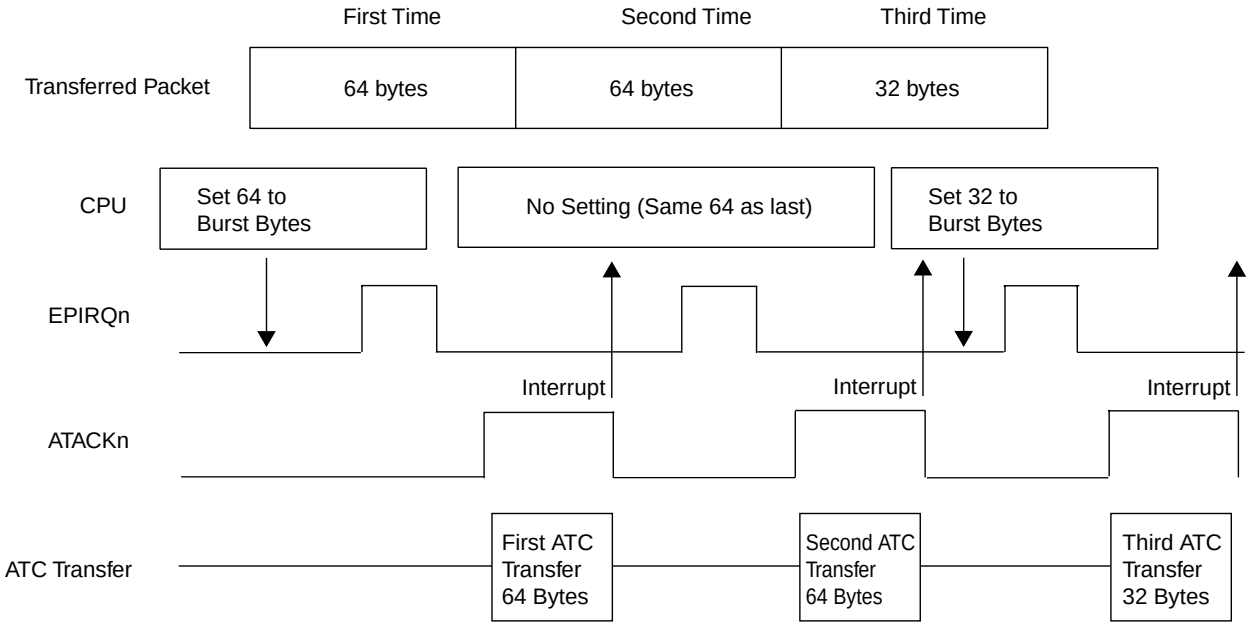


Figure 8-5-7 ATC Transfer Flow for IN Endpoints

When ATC is used for IN endpoints, the CPU must load the first packet to the FIFO. When IN token is sent, if there is no packet in the FIFO, the USB core returns NAK as handshake and does not generate neither EPIRQn nor MC_INTRN. In this condition, no matter how many times the HOST transmits IN token, only NAK is returned and no packet is transmitted. If a packet is loaded to the FIFO at first, after a packet is transmitted, EPIRQn is generated and the next packet is loaded to the FIFO.

■ OUT transfer

When OUT is transferred, a packet is sent from the host. At this time, the packet may be under the size determined by MAXP. To examine the size of sent packet, OWC1 and OWC2 registers need to be referred. When ATC is used, before starting ATC transfer, the size which is burst transferred must be determined with referred OWC1 and OWC2 registers. Figure 8-5-8 shows when endpoint interrupt is generated, the microcontroller service routine needs to be activated. Set burst word setting register of ATC transfer referring OWC1 and OWC2 registers, then start ATC transfer. Transfer all packet data at one ATC transfer. The USB core automatically recognizes that reading packet ends after one ATC transfer ends, and clears MCU_OUT_PKT_RDY.

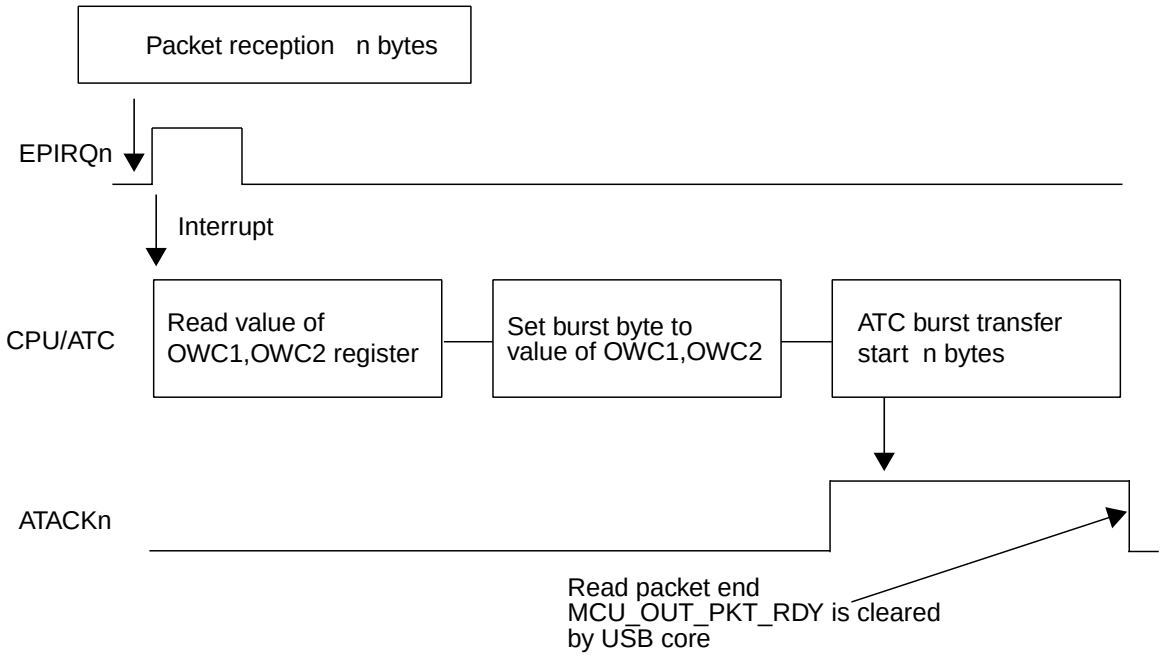


Figure 8-5-8 ATC Transfer for OUT Endpoints

8-6 Programming Considerations

8-6-1 Interrupt Triggers

Table 8-6-1 shows the interrupt triggers available with the interrupt device core, and the bits that trigger them.

Table 8-6-1 Interrupt Triggers

Trigger	Bit Name	Bit No.	Register Name
• USB PROTOCOL			
RESUME	SUSPEND_INTERRUPT set	0	USBI
SUSPEND	RESUME_INTERRUPT set	1	USBI
USB Reset	USB_RESET_INTERRUPT set	2	USBI
• ENDPOINT 0			
Received OUT packet	OUT_PKT_RDY set	0	EP0CSR
Sent IN packet	IN_PKT_RDY cleared	1	EP0CSR
Sent STALL	SENT_STALL set	2	EP0CSR
Control transfer ended and DATA_END successfully	DATA_END cleared	3	EP0CSR
Control transfer ended and DATA_END was not set	SETUP_END set	4	EP0CSR
• IN ENDPOINTS			
Sent IN packet	MCU_IN_PKT_RDY cleared (1)	0	ICSR1
Detected underrun in FIFO	UNDER_RUN set	2	ICSR1
Requested FIFO flush	FIFO_FLUSH cleared	3	ICSR1
Sent STALL	SEND_STALL set	4	ICSR1
• OUT ENDPOINTS			
Received OUT packet	MCU_IN_PKT_RDY set (2)	0	OCSR1
Detected overrun in FIFO	OVER_RUN set	2	OCSR1
Sent STALL	SENT_STALL set	6	CSR1

Notes: When a packet is sent from the FIFO, the MCU_IN_PKT_RDY bit (D0) and the USB_IN_PKT_RDY bit (D1) of the ICSR1 register make either of the following transitions, triggering an interrupt:

two packets in the FIFO in double buffer and transmitted first packet

D0, D1 –> D1, D0

1 1 1 0

one packet in the FIFO in double buffer and transmitted a packet

D1, D0 –> D1, D0

1 0 0 0

one packet in the FIFO in single buffer and transmitted a packet

D1, D0 –> D1, D0

x 1 0 0

Notes: When a packet is received by the FIFO, the MCU_OUT_PKT_RDY bit (D0) and the USB_OUT_PKT_RDY bit (D1) of the OSCR1 register make either of the following transition, triggering an interrupt:

no packets in the FIFO in double buffer and received a packet

D1, D0	–>	D1, D0
0 0		0 1

1 packet in the FIFO in double buffer and received a packet

D1, D0	–>	D1, D0
0 1		1 1

no packets in the FIFO in single buffer and received a packet

D1, D0	–>	D1, D0
x 0		1 1

8-6-2 Interrupt Processing

When interrupt trigger is generated, the USB core writes a one in trigger bit of interrupt register and makes an interrupt line active. Interrupt maintains 'L' until interrupt register (USBI, ENDPI) cleared zero. See figure 8-6-1.

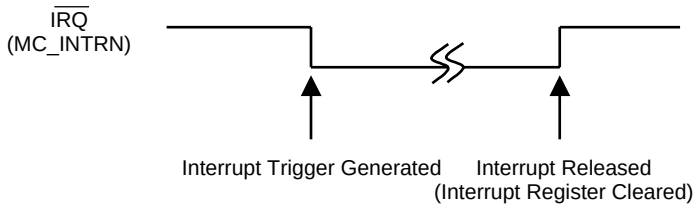


Figure 8-6-1 Interrupt Signal

To release interrupt, write 1 to bit 1 of the interrupt register. By writing the value read from the interrupt register to the interrupt register, the interrupt is released.

> Write in ENDPI register first, then write into USBI register next. Even if the read value of USBI register is zero, write in it.

All interrupt triggers are multiplexed to one IRQ line. Keep in mind when programming that when an interrupt trigger is cleared, there will be other interrupt triggers. The USB core only releases an interrupt once. However, it makes the interrupt line active after one SYSCLK cycle. See figure 8-6-2.

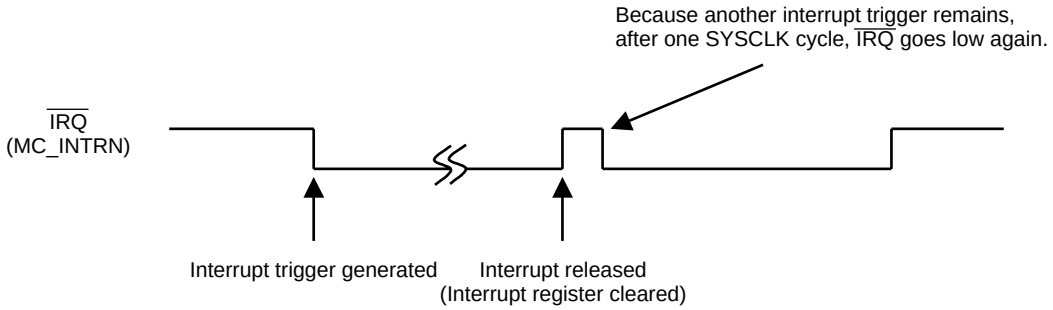


Figure 8-6-2 Interrupt at Remaining Interrupt Triggers 1

At that time, interrupt released timing is synchronized with the operation writing to USBI register. It is generated even if the USBI register value is zero. It is not synchronized with ENDPI register.
When trigger is cleared by ENDPI register, and if other triggers remain, interrupt stays 'L' unless it writes to USBI register. Therefore, interrupt release always needs to write to both USBI and ENDPI registers.

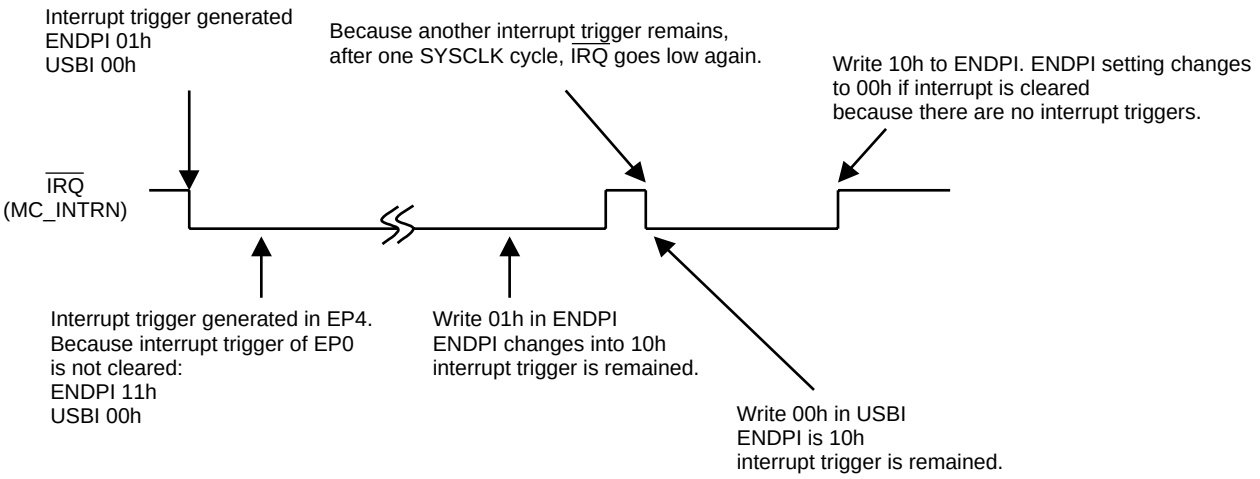


Figure 8-6-3 Interrupt at Remaining Interrupt Triggers 2

To clear interrupt trigger, write in ENDPI register first, then write into USBI register. If 00h is written in USBI when USBI register is 00h into reverse, after one clock, IRQ changes L –> H –> L.

The interrupt is controlled by one bit in each endpoint. In each endpoint, multiple triggers may be generated. If the associated endpoint bit is cleared, all interrupts generated are cleared and no more interrupts will be generated. If long processing time is required, when an interrupt is released, multiple interrupt triggers may be generated for one endpoint. Before an interrupt is released, refer CSR register not to leave any factors.

8-6-3 Processing Multiple Interrupts within the Same USB Frame

Each time the core receives or transmits a data packet for a specific endpoint, it generates an controller interrupt. In accordance with the USB specification, interrupts can only occur once per USB frame for isochronous and interrupt endpoints. With bulk endpoints, however, it is not only possible but common to receive multiple bulk packet interrupts for the same endpoint within a single USB frame. The number of interrupts that the controller can service depends on how fast the controller can process the packets. Optimizing your design for bulk endpoint processing can significantly improve performance speed.

8-6-4 Acknowledging Interrupts from the Core

When the controller receives an interrupt from the core, it reads the interrupt registers to determine the source. It must immediately write back the same read values to clear the interrupt and free the IRQ line for another interrupt. Figure 8-6-4 shows a situation in which a second interrupt occurs before the first is cleared.

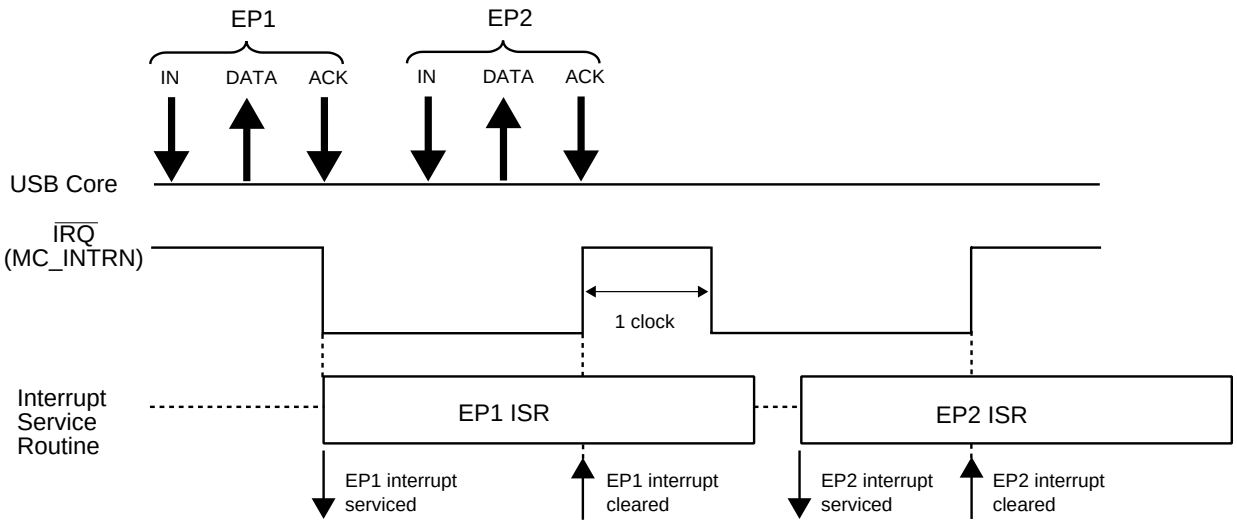


Figure 8-6-4 Interrupt Acknowledgment and Clearing

8-6-5 Servicing Multiple Endpoints within the Same Interrupt

While the controller is servicing an interrupt from one endpoint source, one or more endpoints or events can cause a second, pending interrupt. So that no event is lost, the controller program must handle this situation in one of two ways:

- If a particular event/endpoint requires long processing, the controller can service it outside the interrupt service routine.
- If the controller processes interrupts within the interrupt service routine, the program must check and process all interrupt sources during each interrupt.

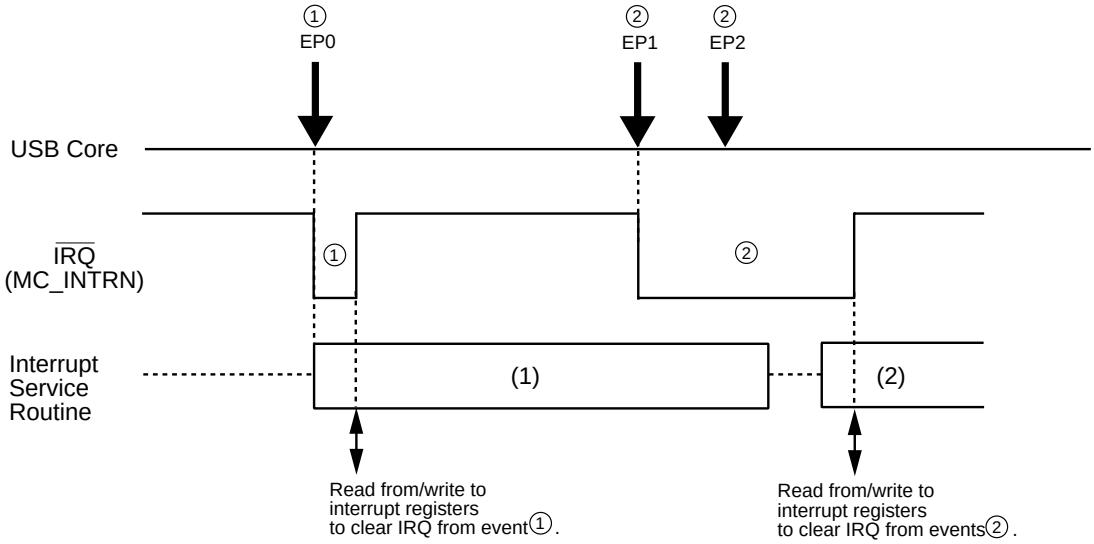


Figure 8-6-5 Multiple Endpoint Requests within the Same Interrupt

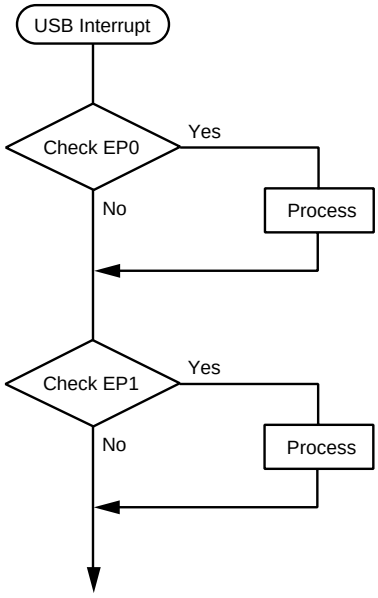


Figure 8-6-6 Example Interrupt Source Checking Flow

8-6-6 Controlling the USB Core for Control Transfers

In this chapter, the operation described as “Clear IRQ and set index” means the following CPU operation will occur.
Read EP_INTERRUPT = 1 USB_INTERRUPT = 0 (Interrupt related with endpoint 0)
Write EP_INTERRUPT = 1 USB_INTERRUPT = 0 –> (Clearing interrupt signal related endpoint 0)
Write INDEX = 0 (Select register in endpoint 0)

(1) **Control Transfers with an IN Direction Data Stage**

Example control transfer with IN data, ending with ACK
Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**
Host: **DATA0 < 80 06 00 01 00 00 12 00 > CRC16 < 072f > (8 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Clear IRQ and set index (1)
Read EP0CSR=01 (endpoint 0 received OUT_PKT_RDY) Read OWC1 Read EP0FIFO (received GET DEVICE DESCRIPTOR) Write 16 bytes to EP0FIFO(loading DEVICE DESCRIPTOR)
Write EP0CSR=43 **(set IN_PKT_RDY, clear OUT_PKT_RDY)**
Host: **IN < ADDR:00 EP:0 > CRC5 <08>**
USB core: **DATA1 < 12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 > CRC16 < 2f6b > (16 Bytes)**
Host: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Clear IRQ and set index (1)
Read EP0CSR=00 (ENDPT0 IN PKT RDY cleared)
Write 2 byte to EP0FIFO (loading DEVICE DESCRIPTOR) Write EP0CSR=0A **(Set IN_PKT_RDY,DATA_END)**
Host: **IN < ADDR:00 EP:0 > CRC5 <08>**
USB core: **DATA0 < 00 01 > CRC16 < fcf1 > (2 Bytes)**
Host: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Clear IRQ and set index (1)
Read EP0CSR=08 (ENDPT0 IN_PKT_RDY cleared, **DATA_END still set**)
Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**
Host: **DATA1 < > CRC16 < 0000 > (0 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index [1]

Read EP0CSR=00 **(ENDPT0 DATA_END cleared)**

Note1: Clear IRQ and set index means:

Read EP_INTERRUPT=1 USB_INTERRUPT=0 Write EP_INTERRUPT=1 USB_INTERRUPT=0

Clear IRQ Write INDEX=0

Example control transfer with IN data when NAK is issued in data phase

Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 80 06 00 01 00 00 12 00 > CRC16 < 072f > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO (Received GET DEVICE DESCRIPTOR)

Do not clear OUT_PKT_RDY in EP0CSR

Host: **IN < ADDR:00 EP:0 > CRC5 <08>**

USB core: **NAK**

MCU: Write 16 byte to EP0FIFO(loading DEVICE DESCRIPTOR)

Write EP0CSR=43 **(Set IN_PKT_RDY, Clear OUT_PKT_RDY)**

Host: **IN < ADDR:00 EP:0 > CRC5 <08>**

USB core: **DATA1 < 12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 > CRC16 < 2f6b > (16 Bytes)**

Host: **ACK**

!!!! IRQ RECEIVED !!!!

The remainder of this example is the same as the previous example.

Example control transfer with IN data when STALL is issued in data phase

Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 80 06 00 01 00 00 12 00 > CRC16 < 072f > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO(Received GET DEVICE DESCRIPTOR)

Write EP0FIFO=61 **(Set SEND_STALL and clear OUT_PKT_RDY)**

Host: **IN < ADDR:00 EP:0 > CRC5 <08>**

USB core: **STALL**

!!!! IRQ RECEIVED !!!!

MCU: **Read EP0CSR=04 (SENT_STALL is set)**

(2) **Control Transfers with an OUT Direction Data Stage**

Example control transfer with OUT data, ending with ACK

Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 42 00 00 00 00 00 1d 00 > CRC16 < ccb2 > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO(Received VENDOR COMMAND)

Write EP0CSR=41**(Clear OUT_PKT_RDY)**

Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA1 < 71 85 4f 3b 3a 7e 15 f1 > CRC16 < a75f > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO

Write EP0CSR=41**(Clear OUT_PKT_RDY)**

Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 86 45 40 86 42 > CRC16 < 2515 > (5 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO(Received VENDOR COMMAND)

Write EP0CSR=49 **(Clear OUT_PKT_RDY and set DATA_END)**

Host: **IN < ADDR:00 EP:0 > CRC5 <08>**

USB core: **DATA1 < > CRC16 < 0000 > (0 Bytes)**

Host: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=10 (DATA_END and OUT_PKT_RDY are cleared)

(2) **Control Transfers with an OUT Direction Data Stage**

Example control transfer with OUT data, ending with ACK

Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 42 00 00 00 00 00 1d 00 > CRC16 < ccb2 > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO(Received VENDOR COMMAND)

Write EP0CSR=41**(Clear OUT_PKT_RDY)**

Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA1 < 71 85 4f 3b 3a 7e 15 f1 > CRC16 < a75f > (8 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO

Write EP0CSR=41**(Clear OUT_PKT_RDY)**

Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**

Host: **DATA0 < 86 45 40 86 42 > CRC16 < 2515 > (5 Bytes)**

USB core: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=01(ENDPT0 Received OUT PKT RDY)

Read OWC1 Read EP0FIFO(Received VENDOR COMMAND)

Write EP0CSR=49 **(Clear OUT_PKT_RDY and set DATA_END)**

Host: **IN < ADDR:00 EP:0 > CRC5 <08>**

USB core: **DATA1 < > CRC16 < 0000 > (0 Bytes)**

Host: **ACK**

!!!! IRQ RECEIVED !!!!

MCU: Clear IRQ and set index

Read EP0CSR=10 (DATA_END and OUT_PKT_RDY are cleared)

8-6-7 Handling Suspend and Resume

When there is no activity on the USB bus, including no SOFs, the USB core detects a suspend status. When this happens, the USB core sets the SUSPEND_INTERRUPT bit (D0) of the USBI register and sends an IRQ to the controller. It also sends the RT_UX_SUSPEND signal to stop the current within the USB transceiver.

> The ENABLE_SUSPEND bit (D0) of the PM register and the SUSPEND_INTERRUPT_EN bit (D0) of the USBIE register must be set for the suspend sequence to work.

> When the bus is in suspend state, it generates an IRQ signal every 3 ms. To avoid unnecessary processing, we recommend that you clear SUSPEND_INTERRUPT_EN once the controller has recognized a suspend sequence.

The controller knows that the host has initiated a suspend by reading the USBI register. When it detects a suspend, the controller must stop the clock to the USB core. If the host initiates a resume signal, the USB core can detect it without clock input. On detecting a resume, the USB core sets the RESUME_INTERRUPT bit (D1) of the USBI register and immediately generates an IRQ to the controller.

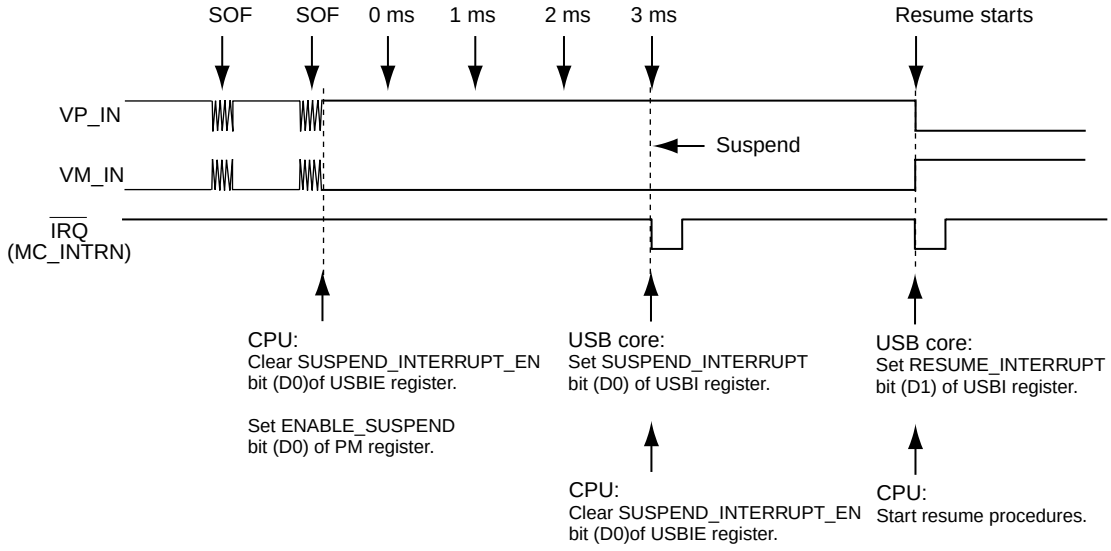


Figure 8-6-7 Suspend Sequence

8-7 Source Code Examples

8-7-1 Short Control Transfer

Host: **SETUP < ADDR:00 EP:0 > CRC5 <08>**
Host: **DATA0 < 80 06 00 01 00 00 12 00 > CRC16 < 072f > (8 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Read EP_INTERRUPT=1 USB_INTERRUPT=0
Write EP_INTERRUPT=1 USB_INTERRUPT=0 —> Clear IRQ
Write INDEX=0
Read EP0CSR=01 (ENDPT0 Received OUT PKT RDY)
Read OWC1
Read EP0FIFO (Received GET DEVICE DESCRIP-TOR)
Write 16 byte to EP0_FIFO (loading DEVICE DESCRIPTOR)
Write EP0CSR=42 (Set IN_PKT_RDY, Clear OUT_PKT_RDY)
Host: **IN < ADDR:00 EP:0 > CRC5 <08>**
USB core: **DATA1 < 12 01 01 00 01 00 00 10 d6 04 00 00 00 00 00 00 > CRC16 < 2f6b > (16 Bytes)**
Host: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Read EP_INTERRUPT=1 USB_INTERRUPT=0
Write EP_INTERRUPT=1 USB_INTERRUPT=0 —> Clear IRQ
Write INDEX=0
Read EP0CSR=00 (IN_PKT_RDY cleared)
Write 2 byte to EP0FIFO (loading DEVICE DESCRIPTOR)
Write EP0CSR=02 **(Set IN_PKT_RDY)**
Host: **OUT < ADDR:00 EP:0 > CRC5 <08>**
Host: **DATA1 < > CRC16 < 0000 > (0 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Read EP_INTERRUPT=1 USB_INTERRUPT=0
Write EP_INTERRUPT=1 USB_INTERRUPT=0 —> Clear IRQ
Write INDEX=0
Read EP0CSR=10 (IN_PKT_RDY cleared, **SETUP_END set)**

8-7-2 OUT Toggle Test on Endpoint 4

Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA0 < 24 81 09 63 0d 8d 65 12 > CRC16 < 0b3a > (8 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: *Read EP_INTERRUPT=10 USB_INTERRUPT=0
Write EP_INTERRUPT=10 USB_INTERRUPT=0 —> Clear IRQ
Write INDEX=4
Read OCSR1=01 (ENDPT4 Received OUT PKT RDY)
Read OWC1,2 Read EP4FIFO (Downloading 8 bytes from EP(4))
Write OCSR1=00 (Clear OUT_PKT_RDY)
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA0 < 01 0d 76 3d ed 8c f9 c6 > CRC16 < 4f77 > (8 Bytes)**
USB core: **ACK**
!!!! NO IRQ !!!!
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA1 < e8 c5 5c bd > CRC16 < c4ec > (4 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA1 < 2d 65 63 0a > CRC16 < 72fa > (4 Bytes)**
USB core: **ACK**
!!!! NO IRQ !!!!
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA0 < 80 20 aa 9d 96 13 0d 53 > CRC16 < df06 > (8 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA0 < 6b d5 02 ae 1d cf 23 0a > CRC16 < dc3f > (8 Bytes)**
USB core: **ACK**
!!!! NO IRQ !!!!
MCU: Write OCSR1=80 (Set CLR_DATA_TOGGLE)
Host: **OUT < ADDR:00 EP:4 > CRC5 <02>**
Host: **DATA0 < ca 3c f2 8a > CRC16 < 2313 > (4 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.

8-7-3 Isochronous OUT Error Test on Endpoint 3

Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < 24 81 09 63 0d 8d 65 12 > CRC16 < 0a3a > (8 Bytes) <ERROR-CRC16!!!**
!!!! IRQ RECEIVED !!!!
MCU: *Read EP_INTERRUPT=08 USB_INTERRUPT=0
Write EP_INTERRUPT=08 USB_INTERRUPT=0 – > Clear IRQ
Write INDEX=3
Read OCSR1=09 (ENDPT3 Received OUT PKT RDY, DATA_ERROR)
Write OCSR1=19 (FLUSHING EP3 FIFO) (1)
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < 01 0d 76 3d ed 8c f9 > CRC16 < a14f > (7 Bytes) <ERROR-CRC16!!!**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < c6 c5 aa e5 77 12 > CRC16 < ad82 > (6 Bytes)**
!!!! IRQ RECEIVED !!!!
MCU: Read EP_INTERRUPT=08 USB_INTERRUPT=0
Write EP_INTERRUPT=08 USB_INTERRUPT=0 –> Clear IRQ
Write INDEX=3
Read OCSR1=01 (ENDPT3 Received OUT PKT RDY)
Read OWC1,2
Read EP3FIFO (Downloading 6 bytes from EP(3))
Write OCSR1=00 (Clear OUT_PKT_RDY)

Note: 1. The application can choose to keep the data rather than flushing it.

8-7-4 Bulk OUT Error Test on Endpoint 3

Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < 24 81 09 63 0d 8d 65 12 > CRC16 < 0a3a > (8 Bytes) <ERROR-CRC16!!!**
!!!! NO HANDSHAKE OR IRQ !!!!
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < 01 0d 76 3d ed 8c f9 > CRC16 < a14f > (7 Bytes) <ERROR-CRC16!!!**
!!!! NO HANDSHAKE OR IRQ !!!!
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < c6 c5 aa e5 77 12 > CRC16 < ad82 > (6 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: *Read EP_INTERRUPT=08 USB_INTERRUPT=0
Write EP_INTERRUPT=08 USB_INTERRUPT=0 –> Clear IRQ
Write INDEX=3
Read OCSR1=01 (ENDPT3 Received OUT PKT RDY)
Read OWC1,2 Read EP3FIFO (Downloading 6 bytes from EP(3)) Write
OCSR1=00 (Clear OUT_PKT_RDY)
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA1 < 8f f2 ce e8 c5 > CRC16 < cb2c > (5 Bytes) <ERROR-CRC16!!!**
!!!! NO HANDSHAKE OR IRQ !!!!
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA1 < 5c bd 2d 65 > CRC16 < 052b > (4 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.
Host: **OUT < ADDR:00 EP:3 > CRC5 <11>**
Host: **DATA0 < 63 0a 80 > CRC16 < 9e84 > (3 Bytes)**
USB core: **ACK**
!!!! IRQ RECEIVED !!!!
MCU: Same as *.

8-8 Internal Registers

The following internal registers are switched depending on endpoints by the INDEX register.

- ICSR1, ICSR2, ICSR3 (IN control status register)
- OCSR1, OCSR2 (Out control status register)
- EP0CSR, EP0CSRX (Endpoint 0 control register)
- MAXP (Packet size register)
- OWC1, OWC2

The USB core has interrupt register and interrupt enable register based on the following interrupt factors.

- In/out endpoint interrupt
- USB interrupt

The MAXP, ENDPI and ENDPIE registers are always used for all endpoints, regardless of the programmed direction. The associated CSR register corresponds to the direction of the endpoints. Note that if none of the endpoints configured as OUT, the OCSR registers do not exist. However, the OUT write count registers exist to read the write count for endpoint zero.

The USB core has two resets. Power-on reset sets all registers to the initial value. Second resets are determined by the USB protocol. This reset is sent from the USB host controller by the USB bus. At this time, some bits of USBIE registers are not reset.

USB Control Registers

Table 8-8-1 List of USB Control Registers

Name	Address	R/W	Function
FA	x'E00510'	R/W	Function Address Register
PM	x'E00512'	R/W	Power Management Register
INDEX	x'E00514'	R/W	Index Register
MAXP	x'E00516'	R/W	MAX Packet Size Register
ENDPI	x'E00518'	R/W	Endpoint Interrupt Register
USBI	x'E0051A'	R/W	USB Interrupt Register
ENDPIE	x'E0051C'	R/W	Endpoint Interrupt Enable Register
USBIE	x'E0051E'	R/W	USB Interrupt Enable Register
SBEP3	x'E00520'	R	Sampling Buffer EP3
SBEP4	x'E00522'	R	Sampling Buffer EP4
FN1	x'E00524'	R	Frame Number Register 1
FN2	x'E00526'	R	Frame Number Register 2
ICSR1(EP0CSR)	x'E00530'	R/W	IN Control Status Register 1
ICSR2	x'E00532'	R/W	IN Control Status Register 2
ICSR3(EP0CSRX)	x'E00534'	R/W	IN Control Status Register 3
OCSR1	x'E00536'	R/W	OUT Control Status Register 1
OCSR2	x'E00538'	R/W	OUT Control Status Register 2
OWC1	x'E0053A'	R	OUT Write Count Register 1
OWC2	x'E0053C'	R	OUT Write Count Register 2
EP0FIFO	x'E00540'	R/W	EP0 FIFO Register
EP1FIFO	x'E00542'	R/W	EP1 FIFO Register
EP2FIFO	x'E00544'	R/W	EP2 FIFO Register
EP3FIFO	x'E00546'	R/W	EP3 FIFO Register
EP4FIFO	x'E00548'	R/W	EP4 FIFO Register
EP5FIFO	x'E0054A'	R/W	EP5 FIFO Register
EP6FIFO	x'E0054C'	R/W	EP6 FIFO Register
EP7FIFO	x'E0054E'	R/W	EP7 FIFO Register
EP8FIFO	x'E00550'	R/W	EP8 FIFO Register
ATCSW	x'E0055E'	R/W	ATC Switch Register

8-8-1 Function Address Register

■ Function address register (FA)

This register maintains the USB device address assigned by the host. This address is used for the next token.

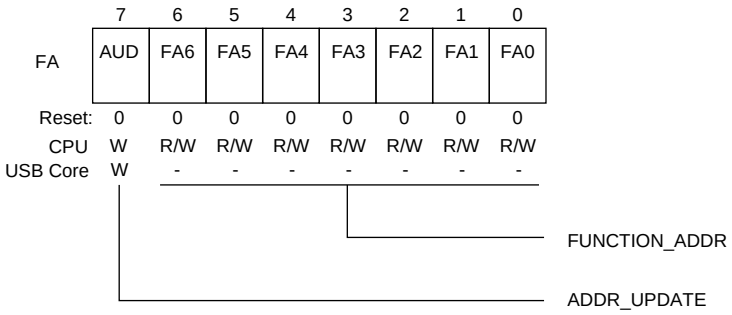


Figure 8-8-1 Function Address Register

Flag Explanation

FUNCTION_ADDR [6 : 0]	Write the device address received from the host.
ADDR_UPDATE	Write '1' when FUNCTION_ADDR is updated. FUNCTION_ADDR is valid after the status phase of a control transfer, which is signaled by the clearing of the DATA_END bit in the EP0CSR register.

8-8-2 Power Management Register

■ Power management register (PM)

This register is used for suspend, resume and reset signaling.

Note: Use the USBI register to poll for suspend and reset conditions. Do not use this register.

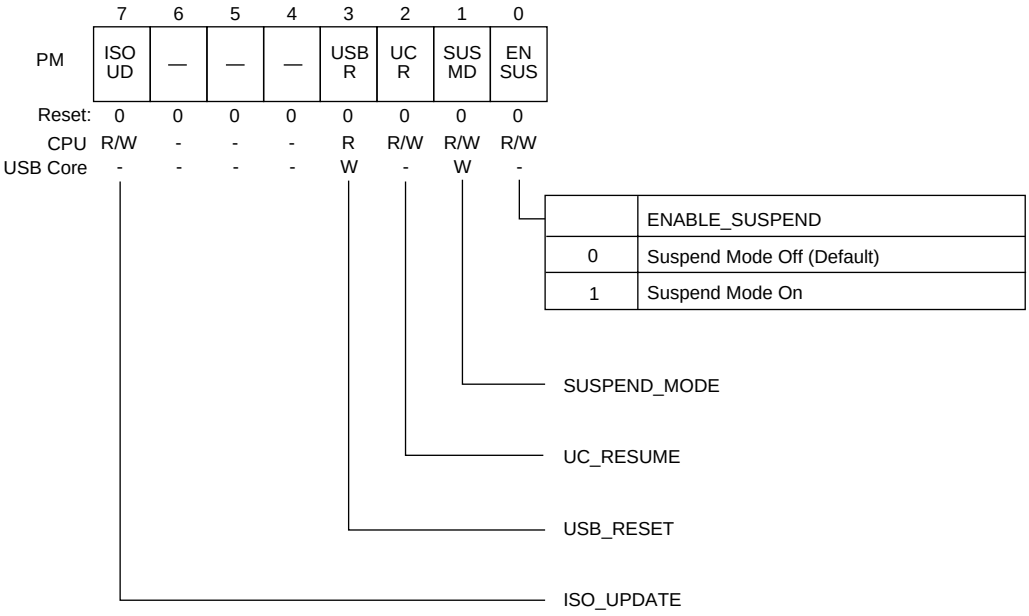


Figure 8-8-2 Power Management Register

Flag Explanation

ENABLE_SUSPEND If this bit is 0, the device will not enter suspend mode.

SUSPEND_MODE The USB core sets this bit when it enters suspend mode. The bit is cleared under the following conditions.

1. The MCU clears the UC_RESUME bit to end resume signaling.
2. The MCU writes '1' to the RESUME_INTERRUPT bit of the USBI register and clear interrupt for USB_RESUME interrupt.

UC_RESUME Sets this bit to 1 for a duration of 10 ms (maximum of 15 ms) to initiate resume signaling. The USB core generates resume signaling while this bit is set in suspend mode.

USB_RESET The USB core sets this bit if reset signaling is received from the host. This bit remains set as long as reset signaling persists on the bus.

ISO_UPDATE This bit is valid only in ISO mode.
If this bit is set, the USB core waits for an SOF token from the time IN_PKT_RDY was set to send the packet. If an IN token is received before an SOF token, then a zero-length data packet will be sent.

8-8-3 Interrupt Registers

The USB core has two interrupt registers, ENDPI and USBI. These registers act as status registers for the controller when an interrupt occurs. The following sections describe each interrupt register. For information on programming interrupt registers, see “8-6 Programming Considerations.”

> To clear a controller interrupt, the controller must read all the interrupt registers and write the read data back to all registers. This write-back operation must be performed irrespective of the read value. The USBI register must be the last register to which it writes.

■ ENDP_Interrupt register (ENDPI)

Each bit in this register corresponds to the respective endpoint number. All interrupts are mapped to this register, regardless of the programmed direction of the endpoint.

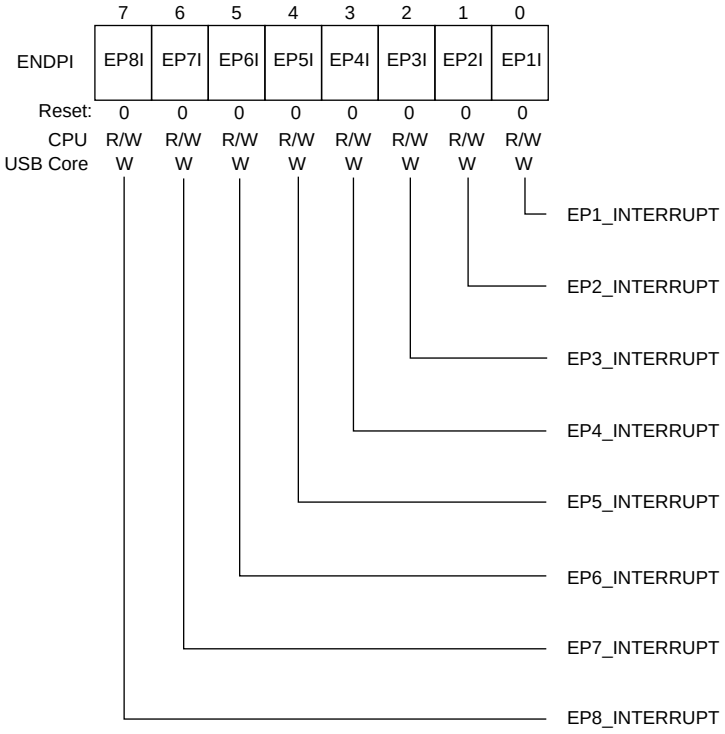


Figure 8-8-3 ENDPI Register

Flag Explanation

EPn_INTERRUPT Each bit is associated with the endpoint having the same number.

For IN, bulk endpoints:
The USB core sets this bit under any of the following conditions.
1. The MCU_IN_PKT_RDY bit (D0) and the USB_IN_PKT_RDY bit (D1) of the ICSR1 register make either of the following transitions, indicating a packet was sent from the FIFO.
2. The FIFO is flushed.

For IN, ISO endpoints:
The USB core sets this bit under the same conditions as those for IN, bulk endpoints. In addition, it sets it when:
1. The UNDER_RUN bit (D2) of the ICSR1 register is set.

For OUT, bulk endpoints:
The USB core sets this bit under either of the following conditions.
1. The MCU_OUT_PKT_RDY bit (D0) and the USB_OUT_PKT_RDY bit (D1) of the OCSR1 register make either of the following transitions, indicating a packet was received by the FIFO.
2. The SENT_STALL bit (D6) of OCSR1 is set.

For OUT, ISO endpoints:
The USB core sets this bit under the same conditions as those for OUT, bulk endpoints. In addition, it sets it when:
1. The OVER_RUN bit (D2) of the OCSR1 register is set.

■ USB_Interrupt register (USBI)

This register maintains interrupt status flags for the following bus signaling conditions:

- Suspend
- Resume
- Reset

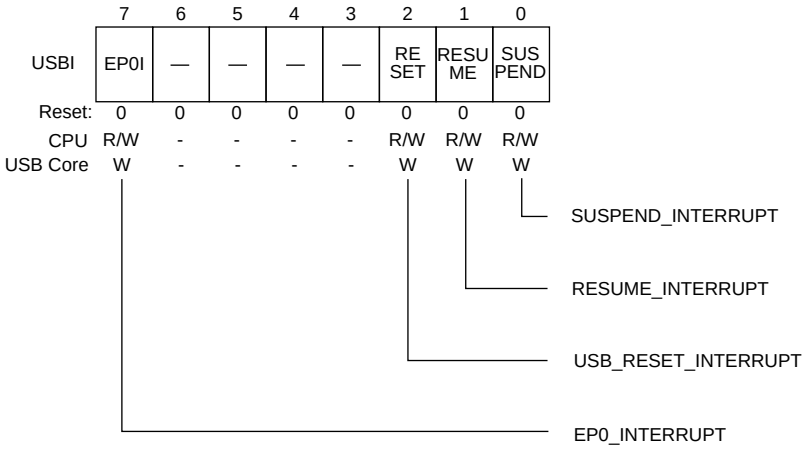


Figure 8-8-4 USBI Register

Flag Explanation

SUSPEND_INTERRUPT — The USB core sets this bit when it receives suspend signaling. This bit is set when ever there is no activity on the bus for 3 ms. Therefore, if the controller does not stop the clock after the first suspend interrupt, it will continue to be interrupted every 3 ms as long as there is no activity on the USB bus. By default this interrupt is disabled.

RESUME_INTERRUPT — The USB core sets this bit when it receives resume signaling while in suspend mode. If the resume is due to a USB reset, the controller is first interrupted with a resume interrupt. Once the clocks resume and the SE0 condition persists for 3 ms, a USB reset interrupt will be asserted.

USB_RESET_INTERRUPT — The USB core sets this bit when it receives reset signaling.
This interrupt resets the USB core registers, so the firmware must re-initialize the registers after it occurs.

EP0_INTERRUPT — This bit corresponds to the endpoint zero. The USB core sets this bit under any of the following conditions.

1. OUT_PKT_RDY (D0) is set.
2. IN_PKT_RDY (D1) is cleared.
3. SENT_STALL (D2) is set.
4. DATA_END (D3) is cleared. (It means the end of the control transition.)
5. SETUP_END (D4) is set.

8-8-4 Interrupt Enable Registers

ENDPIE register and USBIE register controls the endpoint interrupts and USB interrupts. Each number in these registers corresponds to the same bit number in the associated interrupt registers. The interrupt is disable when this bit is set zero.
By default all interrupts except suspend and SOF are enabled.
ENDPIE register is set b'11111111 and USBIE register is set b'10000100 when it resets.

■ ENDP_Interrupt_Enable register (ENDPIE)

This register controls the endpoint interrupts.

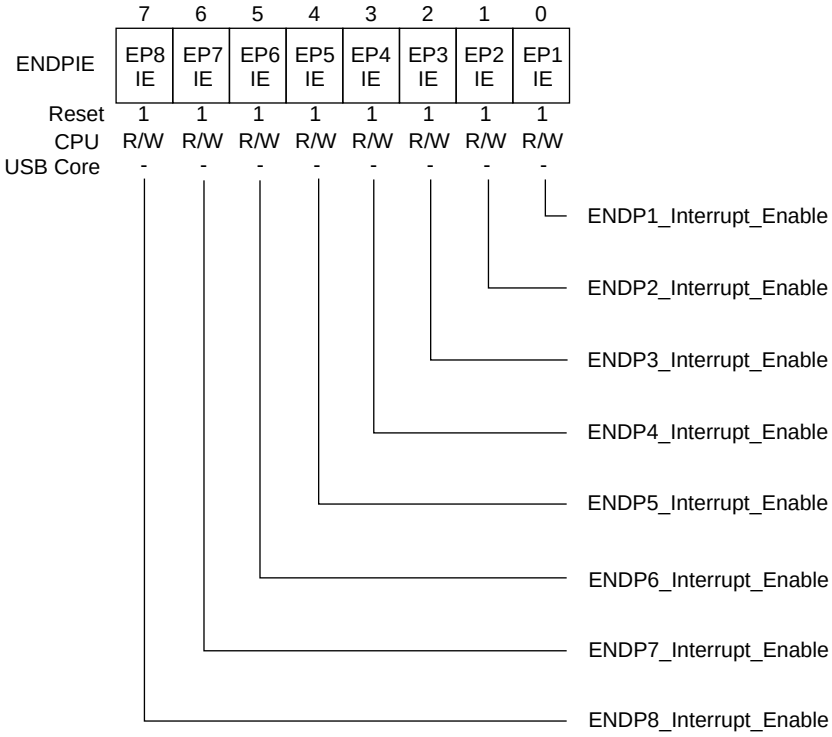


Figure 8-8-5 ENDPIE Register

Flag Explanation

ENDPn_INTERRUPT_ENABLE	Each bit is associated with the endpoint having the same number. 0: Disables this endpoint interrupts 1: Enables this endpoint interrupts

■ USB_Interrupt_Enable register (USBIE)

This register controls the USB interrupts.

	7	6	5	4	3	2	1	0
USBIE	EP0 IE	—	—	—	SOF EN	RESET IE	—	SUS IE
Reset	1	0	0	0	0	1	0	0
CPU	R/W	-	-	-	R/W	R/W	-	R/W
USB Core	-	-	-	-	-	-	-	-

SUSIE	Suspend_Interrupt_Enable
0	Disables Suspend Interrupts
1	Enables Suspend Interrupts

RESETIE	USB_Reset_Interrupt_Enable
0	Disables USB Reset Interrupts
1	Enables USB Reset Interrupts

SOFEN	SOF_Pulse_EN
0	Disables SOF Interrupts
1	Enables SOF Interrupts

EP0IE	EP0_Interrupt_Enable
0	Disables Endpoint 0 Interrupts
1	Enables Endpoint 0 Interrupts

Figure 8-8-6 USBIE Register

Flag Explanation

SUSPEND_INTERRUPT_ENABLE — Suspend interrupts are enabled when this bit is set.
This bit will not be reset, except by power-on reset.

USB_RESET _INTERRUPT_ENABLE — USB interrupts enabled when this bit is set.
This bit will not be reset, except by power-on reset.

SOF_PULSE_ENABLE — SOF interrupt signaling is synchronized with SOF packets and output when this bit is set.
SOF interrupts signaling outputs 1 ms, which is identified in the internal signal standard up to three times if it cannot receive SOF packet from the host. If it cannot receive the fourth SOF_PULSE, the USB core detects suspend and stops SOF interrupts signaling output.

ENDP0_INTERRUPT _ENABLE — Controls endpoint zero interrupts.
0: Disables this endpoint interrupts
1: Enables this endpoint interrupts

8-8-5 Index Register

This register indexes the following registers for each endpoint:

MAXP register
EP0CSR register, EP0CSRX register
ICSR1 register, ICSR2 register, ICSR3 register
OCSR1 register, OCSR2 register
OWC1 register, OWC2 register

■ Index register (INDEX)

	7	6	5	4	3	2	1	0
INDEX	INDEX7	INDEX6	INDEX5	INDEX4	INDEX3	INDEX2	INDEX1	INDEX0
Reset	0	0	0	0	0	0	0	0
CPU	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
USB Core	-	-	-	-	-	-	-	-

INDEX7	INDEX6	INDEX5	INDEX4	INDEX3	INDEX2	INDEX1	INDEX0	ENDPOINT_INDEX
0	0	0	0	0	0	0	0	Endpoint 0
0	0	0	0	0	0	0	1	Endpoint 1
0	0	0	0	0	0	1	0	Endpoint 2
0	0	0	0	0	0	1	1	Endpoint 3
0	0	0	0	0	1	0	0	Endpoint 4
0	0	0	0	0	1	0	1	Endpoint 5
0	0	0	0	0	1	1	0	Endpoint 6
0	0	0	0	0	1	1	1	Endpoint 7
0	0	0	0	1	0	0	0	Endpoint 8

Figure 8-8-7 Index Register

Flag Explanation

ENDPOINT_INDEX Switches over the mapped registers for each endpoint.

8-8-6 Frame Number Registers

These two registers maintain the 11-bit frame number received through the SOF token. The FN1 register maintains the lower 8 bits, and FN2 register maintains the upper three bits.

■ Frame_Number 1 register (FN1)

	7	6	5	4	3	2	1	0
FN1	FN7	FN6	FN5	FN4	FN3	FN2	FN1	FN0
Reset	0	0	0	0	0	0	0	0
CPU	R	R	R	R	R	R	R	R
USB Core	W	W	W	W	W	W	W	W

Figure 8-8-8 Frame_Number1 Register

■ Frame_Number 2 register (FN2)

	7	6	5	4	3	2	1	0
FN2	—	—	—	—	—	FN10	FN9	FN8
Reset	0	0	0	0	0	0	0	0
CPU	-	-	-	-	-	R	R	R
USB Core	-	-	-	-	-	W	W	W

Figure 8-8-9 Frame_Number2 Register

8-8-7 IN Control Status Registers

These registers maintain the configuration and status bits for the IN endpoints. ICSR1 and ICSR3 maintain the status bits, and ICSR2 maintains the configuration bits.

■ IN control status register 1 (ICSR1)

This register maintains the status bits for the IN endpoints. The controller only needs to access this register for an IN endpoint, once the endpoint has been configured. When INDEX register is set 0, the EP0CSR register (described in the following section) will be mapped.

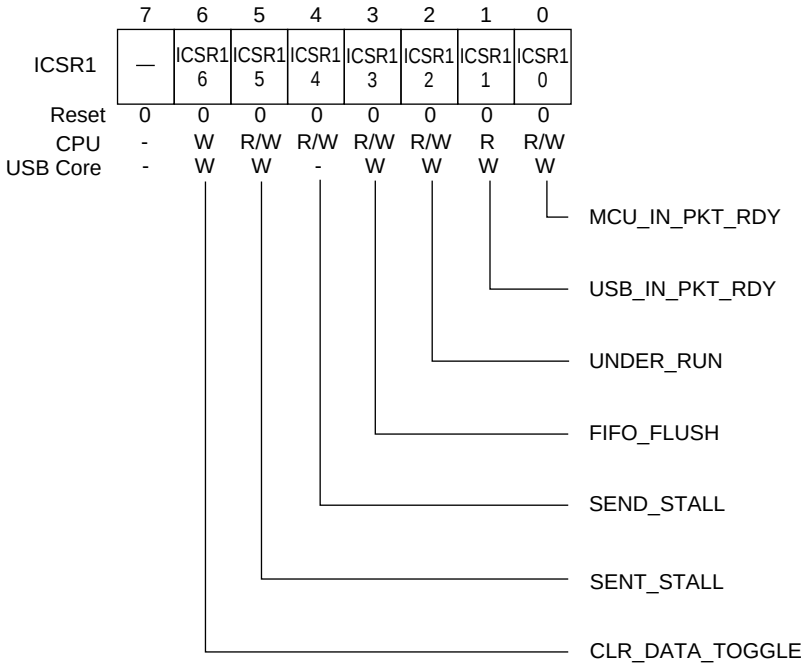


Figure 8-8-10 ICSR1 Register

Flag Explanation

MCU_IN_PKT_RDY Indicates to the controller that it can write a packet of data to the FIFO.
Also, refer AUTO_SET bit.
Set this bit after writing a data packet of data to the FIFO. Since the FIFO can accommodate dual packets, this bit is set only after two packets are loaded into the FIFO. (Note: Dual packets can be accommodated only if MAXP ≤ 1/2.) Therefore when dual packet buffering is enabled, even though the controller writes a 1 to this bit after loading the 1st packet, this bit is cleared immediately and is set only after 2nd packet is loaded into the FIFO and the 1st packet is yet to be transmitted. The USB core clears this bit once the topmost packet in the FIFO has been successfully sent to the host. (For bulk-endpoints, the core will retry a packet if the host does not issue an ACK.)
An interrupt is generated when the USB core clears this bit, so the controller can load the next packet. While this bit is set, the controller will not be able to write to the FIFO. If the SEND_STALL bit is set by the controller, this bit cannot be set.

When double buffering is enabled, the software must set this bit after each packet of data is loaded to the FIFO.

The examples below illustrate correct and incorrect sequencing. In the examples, MAXP = 3 (packet size is 32 bytes).

● Example of correct sequencing:
MCU: Loads 32 bytes
Sets MCU_IN_PKT_RDY
Loads 32 bytes
Sets MCU_IN_PKT_RDY
IN
DATA (32 bytes)
ACK
IN
DATA (32 bytes)
ACK

● Example of incorrect sequencing:
MCU: Loads 64 bytes
Sets MCU_IN_PKT_RDY
Sets MCU_IN_PKT_RDY
IN
DATA (64 bytes)
ACK
IN
DATA (0 bytes)
ACK

Flag Explanation

USB_IN_PKT_RDY Indicates to the USB core that a packet is ready to be transmitted upon receiving an IN token. If an IN token is received and this bit is not set, the core will:
1. Issue a NAK for bulk endpoints
2. Send a zero-length data for ISO endpoints. This USB core sets UNDER_RUN bit and makes an interrupt to the controller.
This bit is set once the controller writes a 1 to MCU_IN_PKT_RDY (D0). This bit is cleared when there are no more packets to be transmitted.

Ignore this bit when forced single buffering is enabled (when SINGLE (D3) of ICSR2 = 1).

USB_IN_PKT_RDY(D1)	MCU_IN_PKT_RDY(D0)	Condition
0	0	No packet in FIFO
0	1	Reserved Setting
1	0	one packet in FIFO with dual buffering
1	1	two packets with dual buffering, or one packet with single buffering (MAXP ≥ FIFO size)

UNDER_RUN Note: This bit is valid only in ISO mode.
The USB core sets this bit during ISO mode when an IN token is received and the MCU_IN_PKT_RDY bit is not set. The USB core sends a zero-length data packet for such conditions, and the next packet that is loaded into the FIFO is flushed.

FIFO_FLUSH The controller writes a 1 to this bit to flush the IN FIFO. The USB core clears it once the FIFO is flushed. The controller is interrupted when this happens. If a token is in progress, the USB core waits until the transmission is complete before the FIFO is flushed. If two packets are loaded into the FIFO, only the topmost packet (the one that was intended to be sent to the host) is flushed, and the corresponding MCU_IN_PKT_RDY bit for that packet is cleared.

SEND_STALL The controller writes a 1 to this bit to issue a STALL handshake to the USB core. The controller clears this bit to end the STALL condition.

SENT_STALL The USB core sets this bit when a STALL handshake is issued to an IN token, when the controller sets the SEND_STALL bit. MCU_IN_PKT_RDY is cleared when the USB core issues a STALL handshake.

CLR_DATA_TOGGLE When the controller writes a 1 to this bit, the data toggle bit is cleared. This is a write-only register.

■ IN control status register 2 (ICSR2)

This register is used to configure IN endpoints.

> Set the MODE bit in this register to configure an OUT endpoint. See section “8-8-8 OUT Control Status Register 2 (OCSR2)” for more information.

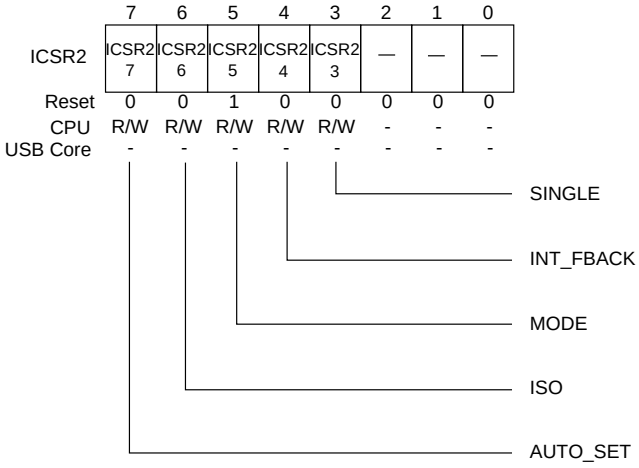


Figure 8-8-11 ICSR2 Register

Flag Explanation

SINGLE 0: Disables forced single buffering (default)
1: Enables forced single buffering

INT_FBACK 0: Disables data rate feedback mode (default)
1: Enables data rate feedback mode

> When this bit is set, the ISO bit must be 0 (bulk / interrupt mode).

MODE 0: Configures endpoint direction as OUT
1: Configures endpoint direction as IN (default)

ISO 0: Configures endpoint to bulk/interrupt mode (default)
1: Configures endpoint to ISO mode

AUTO_SET If this bit is set, whenever the controller writes MAXP data, the core automatically sets MCU_IN_PKT_RDY, without any intervention from the controller. If the controller writes less than MAXP data, then the controller must set the MCU_IN_PKT_RDY. Default = 0

> When writing to AUTO_SET, your program must preserve the settings in the MODE (D5) and ISO (D6) bits.

■ IN control status register 3 (ICSR3)

This register maintains status bits for the IN endpoints. When INDEX register sets zero, the EP0CSRX register (described in the following section) will be mapped.

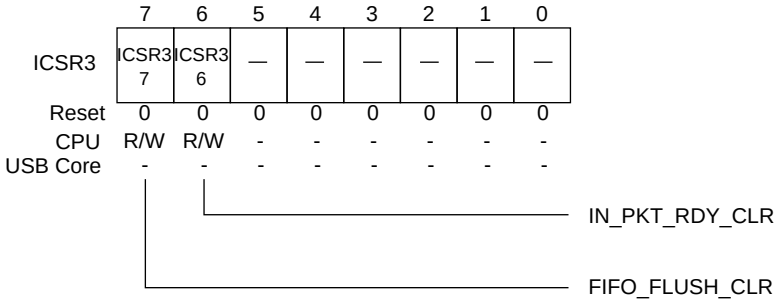


Figure 8-8-12 ICSR3 Register

Flag Explanation

IN_PKT_RDY_CLR — The USB core sets this bit when it clears either the MCU_IN_PKT_RDY or the USB_IN_PKT_RDY bit of the ICSR1 register.

FIFO_FLUSH_CLR — The USB core sets this bit when it clears the FIFO_FLUSH bit of the ICSR1 register.

8-8-8 OUT Control Status Registers

The controller uses OCSR1 and OCSR2 to control OUT endpoints. OCSR1 maintains status information, while OCSR2 is used to configure the endpoint.

■ OUT control status register 1 (OCSR1)

This register maintains the status bits for the OUT endpoints. The controller only needs to access this register for an OUT endpoint, once the endpoint has been configured.

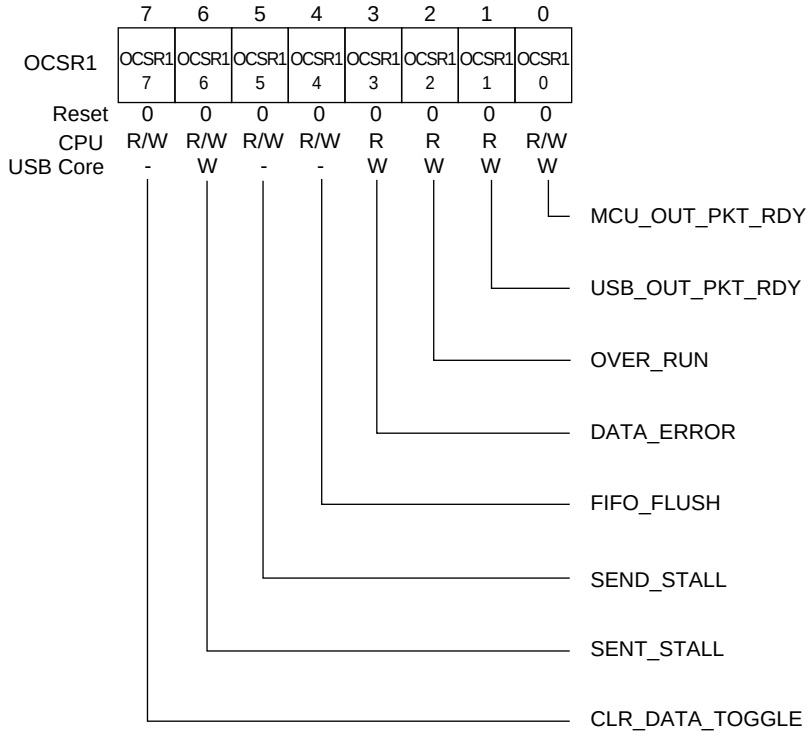


Figure 8-8-13 OCSR1 Register

Flag Explanation

MCU_OUT_PKT_RDY This bit indicates to the controller that at least one packet of data needs to be unloaded. The USB core sets this bit once it has loaded a packet of data into the FIFO. Once the controller reads the FIFO for the entire packet, the controller should clear this bit by writing a 0 to it. (See the description of the AUTO_CLR bit.) If there is another packet in the FIFO (due to dual packet buffering), then this bit is set immediately.

USB_OUT_PKT_RDY Indicates to the USB core that the FIFO is full. While this bit is set, the USB core:

- For bulk endpoints: Issues a NAK to OUT data packets
- For ISO endpoints: Stops loading the FIFO and sets the OVER_RUN bit (D2)

Ignore the bit when forced single buffering is enabled (when SINGLE (D3) of OUT_CSR2 = 1).

USB_OUT_PKT_RDY(D1)	MCU_OUT_PKT_RDY(D0)	Condition
0	0	No packets in FIFO
0	1	one packet in FIFO
1	0	Reserved setting
1	1	two packets with dual buffering, or one packet with single buffering (MAXP ≥ FIFO size)

OVER_RUN Note: This bit is valid only in ISO mode.
This bit is set if the core is unable to load an OUT ISO token into the FIFO.

DATA_ERROR Note: This bit is valid only in ISO mode.
This bit should be sampled with MCU_OUT_PKT_RDY (D0). When set, this bit indicates that the data packet due to be unloaded by the controller has an error (either bit stuffing or CRC). If two packets are loaded into the FIFO, and the 2nd packet has an error, then this bit is set only after the first packet is unloaded. This bit is automatically cleared when MCU_OUT_PKT_RDY is cleared.

FIFO_FLUSH The controller writes a 1 to this bit to flush the FIFO. This bit can be set only when controller_OUT_PKT_RDY (D0) is set. The packet due to be unloaded by the controller will be flushed. Write two times to flush two packets.

SEND_STALL The controller writes a 1 to this bit to issue a STALL handshake to the USB core. The controller clears this bit to end the STALL condition.

SENT_STALL The USB core sets this bit when an OUT token is ended with a STALL handshake. The USB core issues a STALL handshake to the host if it sends more than MAXP data for the OUT token.

CLR_DATA_TOGGLE When the controller writes a 1 to this bit, the data toggle sequence bit is reset to DATA0.

■ OUT control status register 2 (OCSR2)

This register is used to configure IN endpoints.

> To configure an endpoint as OUT, set the MODE bit (D5) of the ICSR2 register to 0.

For example, to setup an OUT, ISO endpoint:

1. Set MODE in ISCR2 to 0.
2. Set ISO in OCSR2 to 1.

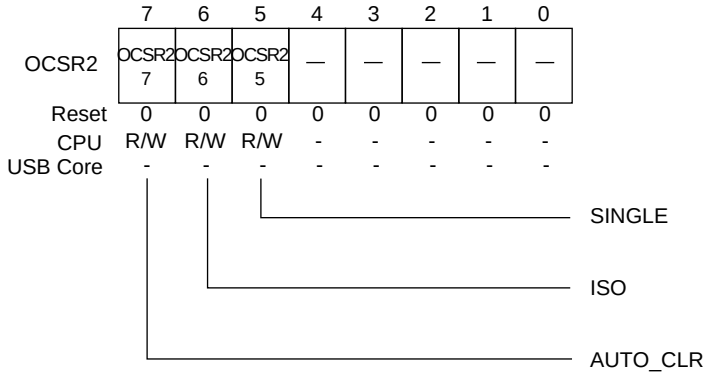


Figure 8-8-14 OCSR2 Register

Flag Explanation

SINGLE 0: Disables forced single buffering (default)
1: Enables forced single buffering

ISO 0: Configures endpoint to bulk/interrupt mode (default)
1: Configures endpoint to ISO mode

AUTO_CLR If this bit is set, whenever the controller reads data from the OUT FIFO, MCU_OUT_PKT_RDY will automatically be cleared by the core, without any intervention from the controller. Default = 0

8-8-9 Endpoint 0 Control Status Register

This register has the control and status bits for endpoint zero. Since a control transaction involves both IN and OUT tokens, there is only one CSR register, mapped to the ICSR1 register when the INDEX register is zero.

■ Endpoint 0 control status register (EP0CSR)

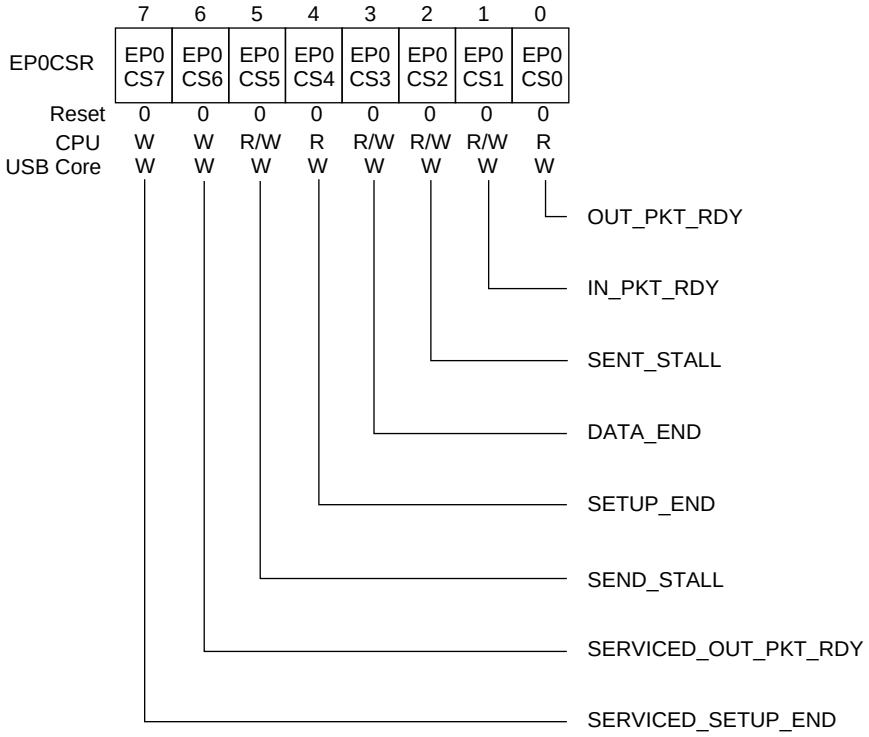


Figure 8-8-15 EP0CSR Register

Flag Explanation

OUT_PKT_RDY — This is a read-only bit. The USB core sets this bit once a valid token is written to the FIFO. An interrupt is generated when the USB core sets this bit. The controller clears this bit by writing a 1 to the SERVICED_OUT_PKT_RDY bit (D6).

> Endpoint 0 does not provide dual packet buffering. Therefore there are no separate IN(OUT)_PKT_RDY bits for the controller and the USB core.

IN_PKT_RDY — The controller sets this bit after writing a packet of data into the endpoint zero FIFO. The USB core clears this bit once the packet has been successfully sent to the host. An interrupt is generated when the USB core clears this bit, so the controller can load the next packet. For a zero-length data phase, the controller sets IN_PKT_RDY and DATA_END (D3) at the same time.

SENT_STALL — The USB core sets this bit if a control transaction is ended due to a protocol violation. An interrupt is generated when this bit is set.

DATA_END — The controller sets this bit:
1. After loading the last packet of data into the FIFO, at the same time IN_PKT_RDY is set.
2. While it clears OUT_PKT_RDY after unloading the last packet of data.
3. For a zero-length data phase, when it clears OUT_PKT_RDY and sets IN_PKT_RDY.

SETUP_END — When a control transfer ends, the USB core sets this bit, generates an interrupt to the controller, and flushes the FIFO. The controller clears this bit by writing a 1 to the SEVICED_SETUP_END (D7) bit.

SEND_STALL — The controller writes a 1 to this bit at the same time that it clears OUT_PKT_RDY (D0), if an invalid device request is received or when the backend application cannot process the device request. The USB core issues a STALL handshake to the current control transfer. The controller writes a zero to end the STALL condition.

SERVICED_OUT_PKT_RDY — The controller writes a 1 to this bit to clear OUT_PKT_RDY. This is a write-only bit.

SERVICED_SETUP_END — The controller writes a 1 to this bit to clear SETUP_END. This is a write-only bit.

> In control transfers in which there is no data phase, the controller sets IN_PKT_RDY after unloading the setup token and sets DATA_END at the same time that it clears OUT_PKT_RDY for the setup token.

> When the SETUP_END bit is set, the OUT_PKT_RDY bit can also be set. This happens when the current transfer has ended and a new control transfer is received before the controller can service the interrupts. In such a case, the controller should first clear the SETUP_END bit, then start servicing the new control transfer.

■ Endpoint 0 control status X register (EP0CSRX)

This register is mapped to the ICSR3 register when the Index register is set to zero.

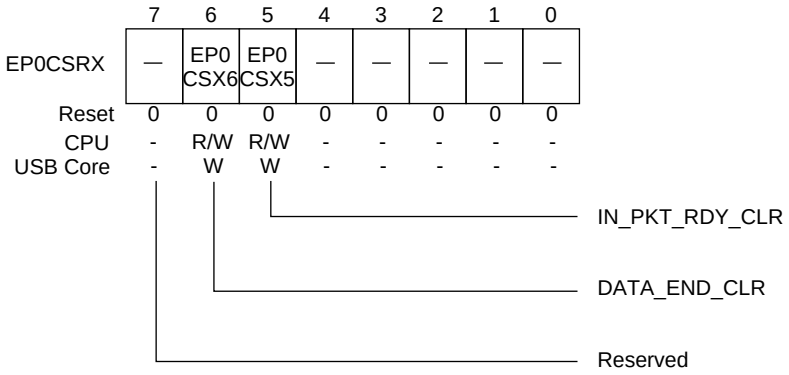


Figure 8-8-16 EP0CSRX Register

Flag Explanation

IN_PKT_RDY_CLR The USB core sets this bit when it clears the IN_PKT_RDY bit of the EP0CSR register.

DATA_END_CLR The USB core sets this bit when it clears the DATA_END bit of the EP0CSR register.

Reserved This bit must be set to zero.

8-8-10 Maximum Packet Size Register

This register holds the maximum packet size for all endpoints. The packet size setting is varied in multiples of eight bytes. If the controller writes a value greater than the FIFO size (for example, 128 bytes for endpoint zero), the MAXP register will maintain the maximum FIFO size.

■ MAX packet size register (MAXP)

	7	6	5	4	3	2	1	0
MAXP	MAXP 7	MAXP 6	MAXP 5	MAXP 4	MAXP 3	MAXP 2	MAXP 1	MAXP 0
Reset	0	0	0	0	0	0	0	1
CPU	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
USB Core	-	-	-	-	-	-	-	-

MAXP7	MAXP6	MAXP5	MAXP4	MAXP3	MAXP2	MAXP1	MAXP0	FIFO Packet Size Setting Flag
0	0	0	0	0	0	0	0	MAXP=0
0	0	0	0	0	0	0	1	MAXP=8 (Default)
0	0	0	0	0	0	1	0	MAXP=16
0	0	0	0	0	1	0	0	MAXP=32
0	0	0	0	1	0	0	0	MAXP=64
0	0	0	1	0	0	0	0	MAXP=128
00010001-11111111								Reserved (Setting Disable)

Figure 8-8-17 MAXP Register

The value in the MAXP register must be the same as the device descriptor value.

8-8-11 OUT Write Count Registers

There are two registers, OWC1 register and OWC2 register, which maintain the write count. OWC1 maintains the lower byte, while OWC2 maintains the higher byte. When MCU_OUT_PKT_RDY is set for OUT endpoints, these registers maintain the number of bytes in the packet due to be unloaded by the controller.

■ OUT write count register 1 (OWC1)

	7	6	5	4	3	2	1	0
OWC1	OWC 7	OWC 6	OWC 5	OWC 4	OWC 3	OWC 2	OWC 1	OWC 0
Reset	0	0	0	0	0	0	0	0
CPU	R	R	R	R	R	R	R	R
USB Core	W	W	W	W	W	W	W	W

Figure 8-8-18 OWC1 Register

■ OUT write count register 2 (OWC2)

	7	6	5	4	3	2	1	0
OWC2	OWC 15	OWC 14	OWC 13	OWC 12	OWC 11	OWC 10	OWC 9	OWC 8
Reset	0	0	0	0	0	0	0	0
CPU	R	R	R	R	R	R	R	R
USB Core	W	W	W	W	W	W	W	W

Figure 8-8-19 OWC2 Register

8-8-12 Sampling Buffers

SBEP3 and SBEP4 registers capture the last byte of transferred data from each transferred packet using endpoint 3 or endpoint 4. These registers were created for the CCD camera application, to allow data sampling.

■ Sampling buffer (SBEP3)

	7	6	5	4	3	2	1	0
SBEP3	SB37	SB36	SB35	SB34	SB33	SB32	SB31	SB30
Reset	0	0	0	0	0	0	0	0
CPU	R	R	R	R	R	R	R	R
USB Core	W	W	W	W	W	W	W	W

Figure 8-8-20 Sampling Buffer EP3

■ Sampling buffer (SBEP4)

	7	6	5	4	3	2	1	0
SBEP4	SB47	SB46	SB45	SB44	SB43	SB42	SB41	SB40
Reset	0	0	0	0	0	0	0	0
CPU	R	R	R	R	R	R	R	R
USB Core	W	W	W	W	W	W	W	W

Figure 8-8-21 Sampling Buffer EP4

8-8-13 ATC Switch Register

This register controls the expansion of interrupt triggers in endpoints one to eight.

■ ATC switch register (ATCSW)

When this register sets 1 to each bit, each endpoint interrupt signal (EPIRQn (n = 1-8)) is generated.
See “8-5 ATC Interface”.

1. IN: When transmission of packet ends and IN_RKT_RDY is cleared, interrupt is generated in the associated endpoints.
2. OUT: When reception of packets ends and OUT_RKT_RDY is set, interrupt is generated in the associated endpoints.
   When endpoint interrupt is generated, USB general interrupt is not generated.

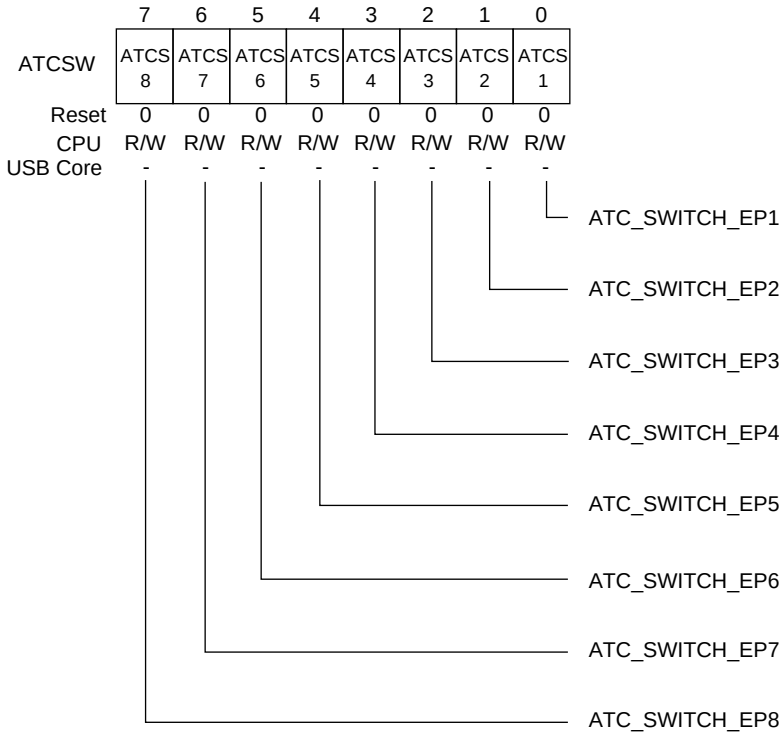


Figure 8-8-22 ATC_SWITCH Register

> When ATCSW register is not used, USB general interrupt is generated in each endpoint. In that case, It needs specification of factor by reading ENDPI register (8-8-3) in the interrupt routine.

Chapter 9 DMA

9-1 Summary of ATC Function

9-1-1 ATC Function

The ATC (Automatic Transfer Control) transfers the data between internal peripheral function/ internal memory and external device/external memory automatically. The ATC do not need to transfer the data throughout the CPU core.

Table 9-1-1 ATC Functions

Transfer Channel	Four channels (ATC 0 to ATC 3) Transfer priority changeable
Address Mode Note 1	Single address mode (only between USB-FIFO and internal RAM) Dual address mode (all spaces)
Wait Control Mode	Fixed wait mode Handshake mode
Bus Mode Note 2	C-bus mode early read/write mode 1-cycle mode
Transfer Mode	1-word transfer Burst transfer
Transfer Unit	Byte Word
Address Specification	+1, +2, fixed, -1, -2
Activation	Software activation Hardware activation
External Bus Request	Arbitrate between external bus request and ATC transfer request Priority order: external bus request > ATC0 to ATC3

Note 1: The single address mode operates read and write at the same time during one bus transaction.
Note 2: Set the bus mode independently although the CPU sets the bus mode for each space.

The space for x'008000' to x'00FFFF' (reserve space and internal I/O control register (1)) is not able to access by ATC.

In single-chip mode and memory expansion mode, internal ROM space (x'080000' to x'1FFFFF') is not able to access by ATC.

9-1-2 ATC Transfer Block Diagram

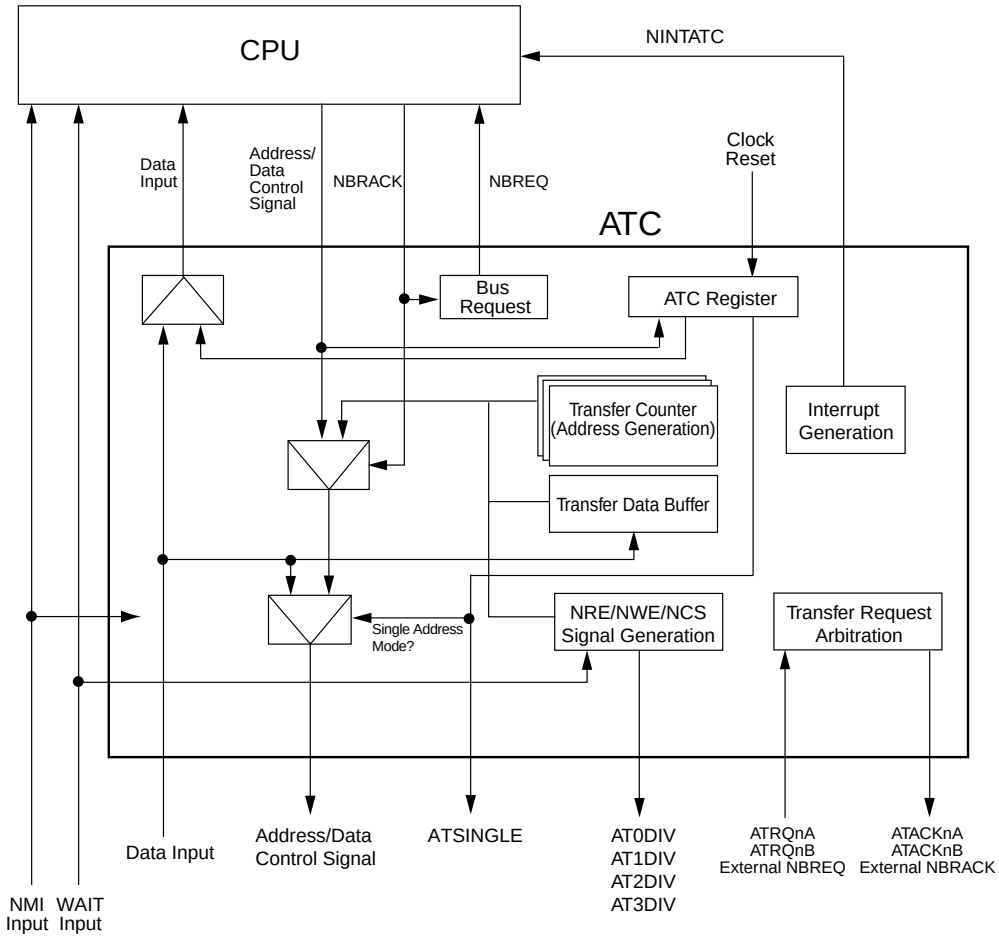


Figure 9-1-1 ATC Block Diagram

9-2 ATC Control Registers

These registers control the ATC; the ATC control register (ATnCTR), the ATC memory mode register (ATnMMD, ATnMMDS), the ATC interrupt register (ATIRQ), the ATC transfer word register (ATnCNT), the ATC source address register (ATnSRC), the ATC destination address register (ATnDST), the ATC transfer channel priority setup register (ATPRI), the ATC input channel select register (ATSEL) and the ATC hardware activation factor select register (ATRQSEL).

Table 9-2-1 List of ATC Control Registers

Register	Address	R/W	Function
AT0CTR	x'E00400'	R/W	ATC0 Control Register
AT0MMD	x'E00410'	R/W	ATC0 Memory Mode Register
AT0MMDS	x'E00418'	R/W	ATC0 Memory Mode Register
AT0CNT	x'E00430'	R/W	ATC0 Transfer Word Register
AT0SRC	x'E00440'	R/W	ATC0 Source Address Register
AT0DST	x'E00450'	R/W	ATC0 Destination Address Register
AT1CTR	x'E00402'	R/W	ATC1 Control Register
AT1MMD	x'E00412'	R/W	ATC1 Memory Mode Register
AT1MMDS	x'E0041A'	R/W	ATC1 Memory Mode Register
AT1CNT	x'E00432'	R/W	ATC1 Transfer Word Register
AT1SRC	x'E00444'	R/W	ATC1 Source Address Register
AT1DST	x'E00454'	R/W	ATC1 Destination Address Register
AT2CTR	x'E00404'	R/W	ATC2 Control Register
AT2MMD	x'E00414'	R/W	ATC2 Memory Mode Register
AT2MMDS	x'E0041C'	R/W	ATC2 Memory Mode Register
AT2CNT	x'E00434'	R/W	ATC2 Transfer Word Register
AT2SRC	x'E00448'	R/W	ATC2 Source Address Register
AT2DST	x'E00458'	R/W	ATC2 Destination Address Register
AT3CTR	x'E00406'	R/W	ATC3 Control Register
AT3MMD	x'E00416'	R/W	ATC3 Memory Mode Register
AT3MMDS	x'E0041E'	R/W	ATC3 Memory Mode Register
AT3CNT	x'E00436'	R/W	ATC3 Transfer Word Register
AT3SRC	x'E0044C'	R/W	ATC3 Source Address Register
AT3DST	x'E0045C'	R/W	ATC3 Destination Address Register
ATIRQ	x'E00420'	R/W	ATC Interrupt Register
ATPRI	x'E00422'	R/W	ATC Transfer Channel Priority Setup Register
ATSEL	x'E00424'	R/W	ATC Input/Output Channel Select Register
ATRQSEL	x'E00460'	R/W	ATC Hardware Activation Factor Select Register

■ ATC Control Register (ATnCTR) (n=0 to 3)

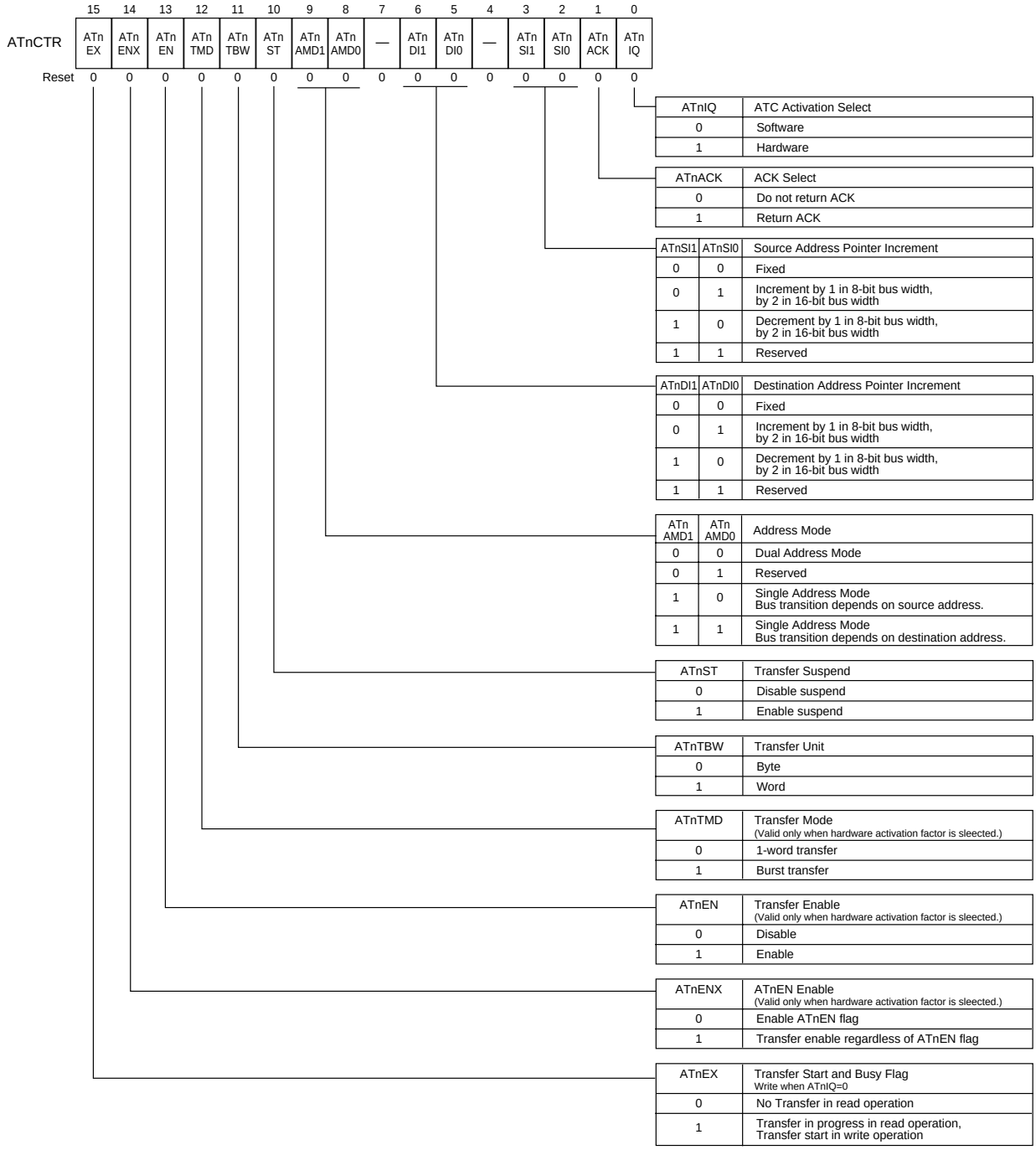

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ATnCTR
ATn EX
ATn ENX
ATn EN
ATn TMD
ATn TBW
ATn ST
ATn AMD1
ATn AMD0
—
ATn DI1
ATn DI0
—
ATn SI1
ATn SI0
ATn ACK
ATn IQ
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ATnIQ | ATC Activation Select
0 | Software
1 | Hardware
ATnACK | ACK Select
0 | Do not return ACK
1 | Return ACK
ATnSI1 | ATnSI0 | Source Address Pointer Increment
0 | 0 | Fixed
0 | 1 | Increment by 1 in 8-bit bus width, by 2 in 16-bit bus width
1 | 0 | Decrement by 1 in 8-bit bus width, by 2 in 16-bit bus width
1 | 1 | Reserved
ATnDI1 | ATnDI0 | Destination Address Pointer Increment
0 | 0 | Fixed
0 | 1 | Increment by 1 in 8-bit bus width, by 2 in 16-bit bus width
1 | 0 | Decrement by 1 in 8-bit bus width, by 2 in 16-bit bus width
1 | 1 | Reserved
ATn AMD1 | ATn AMD0 | Address Mode
0 | 0 | Dual Address Mode
0 | 1 | Reserved
1 | 0 | Single Address Mode Bus transition depends on source address.
1 | 1 | Single Address Mode Bus transition depends on destination address.
ATnST | Transfer Suspend
0 | Disable suspend
1 | Enable suspend
ATnTBW | Transfer Unit
0 | Byte
1 | Word
ATnTMD | Transfer Mode (Valid only when hardware activation factor is sleected.)
0 | 1-word transfer
1 | Burst transfer
ATnEN | Transfer Enable (Valid only when hardware activation factor is sleected.)
0 | Disable
1 | Enable
ATnENX | ATnEN Enable (Valid only when hardware activation factor is sleected.)
0 | Enable ATnEN flag
1 | Transfer enable regardless of ATnEN flag
ATnEX | Transfer Start and Busy Flag Write when ATnIQ=0
0 | No Transfer in read operation
1 | Transfer in progress in read operation, Transfer start in write operation

Figure 9-2-1 ATC Control Register (Word Access only, R/W)

[Flag Explanation]

ATnIQ — Select hardware activation factor (ATRQnA signal, ATRQnB signal) or software activation factor (ATnEX flag setup).

ATnACK — Select ACK return or not by ATACKn pin. Set this flag when recognizing DMA transfer in progress on device regardless of hardware activation or software activation. Set this flag to 1 in USB-FIFO single address mode.

ATnSI[1:0] — Specify whether the ATnSRC register is increment (by 1 or 2), decrement (by 1 or 2) or fixed. The increment value or decrement value is set by the ATnBW flag of the ATnMMD register.

When ATnBW=0 (16-bit bus width), increment or decrement by 2
When ATnBW=1 (8-bit bus width), increment or decrement by 1

ATnDI[1:0] — Specify whether the ATnDST register is increment (by 1 or 2), decrement (by 1 or 2) or fixed. The increment value or decrement value is set by the ATnBW flag of the ATnMMD register.

When ATnBW=0 (16-bit bus width), increment or decrement by 2
When ATnBW=1 (8-bit bus width), increment or decrement by 1

ATnAMD[1:0] — Specify the address mode. When the single address mode is selected, the source address or the destination address should be set based on the addressing in the internal RAM.

ATnST — Show whether a NMI interrupt causes the transfer suspend or not. The ATnST flag becomes 1 when the transfer suspends. Writing only 0 to the ATnST flag is allowed. Restarting the transfer by software activation clears the ATnST flag.

ATnTBW — Specify the transfer unit. Selecting the word transfer as transfer unit cannot be allowed when both the source address and the destination address are set to 8-bit bus space.

When both source address and destination address are on 16-bit bus space, select byte or word unit.
When either source address or destination address is on 16-bit bus space and another is on 8-bit bus space, select byte or word unit. (When word unit is selected, the access cycle on 8-bit bus space is activated twice.)
When both source address and destination address are on 8-bit bus space, select byte unit.

ATnTMD — Specify the burst transfer or 1-word transfer when hardware activation factor is selected. Select the burst transfer whenever software activation factor is selected.

ATnEN — Enable or disable the transfer. Valid the ATnEN flag only when hardware activation factor is selected. **This flag determines whether a bus request is sent to the CPU or not. Therefore, ATC accepts a bus request from the peripheral device even though the ATnEN flag is disabled, and then ATC sends a bus request to the CPU when the ATnEN flag is enabled. Changing the ATnIQ flag from 1 to 0 clears the bus request from the peripheral device.** This flag is automatically reset (transfer disable) after the transfer is completed.

ATnENX — Enable the transfer regardless of the ATnEN value. Valid the ATnENX flag only when hardware activation factor is selected. Difference between the ATnEN flag and the ATnENX flag is disabling the transfer after transfer completion or enabling continuous transfer. Set the ATnENX to 1 when setting for the next transfer and setting the ATnEN flag to 1 cannot be made in time during the interrupt service routine.

ATnEX

This flag function is different depending on the read operation or the write operation.

Write: this flag is set only when software activation factor is selected. Setting this flag to 1 enables a bus request. Setting this flag to 0 during 1-word transfer disables the transfer.
Read: this flag identifies whether the transfer is in progress or the transfer is halted. This flag is always 0 when the burst transfer mode is selected. (ATC does not operate during CPU operation.) In the 1-word transfer mode, this flag becomes 1 when ATC starts transferring the first word and becomes 0 when ATC completes the sequence of transfers.

■ ATC Memory Mode Register (ATnMMD, ATnMMDS)

The ATnMMD register or the ATnMMDS register shows the data transfer mode to transfer the data using ATC, while the memory mode register (MEMMDn, MEMMDnS) on the CPU core selects the mode to transfer the data using CPU. The following table shows the valid memory space for each register. Setting the ATnDIV flag (the eleventh bit of the ATnMMD register) divides the memory space into two areas as follows. Therefore, these ATnMMD registers and ATnMMDS registers can be allocated on up to eight areas by setting the ATnDIV flag to 1.

Table 9-2-2 ATC Memory Space

AT0DIV = 0	AT0MMD	x'000000' to x'3FFFFF'
AT0DIV = 1	AT0MMD AT0MMDS	x'000000' to x'1FFFFF' x'200000' to x'3FFFFF'
AT1DIV = 0	AT1MMD	x'400000' to x'7FFFFF'
AT1DIV = 1	AT1MMD AT1MMDS	x'400000' to x'5FFFFF' x'600000' to x'7FFFFF'
AT2DIV = 0	AT2MMD	x'800000' to x'BFFFFF'
AT2DIV = 1	AT2MMD AT2MMDS	x'800000' to x'9FFFFF' x'A00000' to x'BFFFFF'
AT3DIV = 0	AT3MMD	x'C00000' to x'FFFFFF'
AT3DIV = 1	AT3MMD AT3MMDS	x'C00000' to x'DFFFFF' x'E00000' to x'FFFFFF'

Take notice that the distinction of ATC memory mode register (ATnMMD, ATnMMDS) means the partitioning of the memory space. It differs from the distinction such as ATC transfer channel ATC0 to 3.

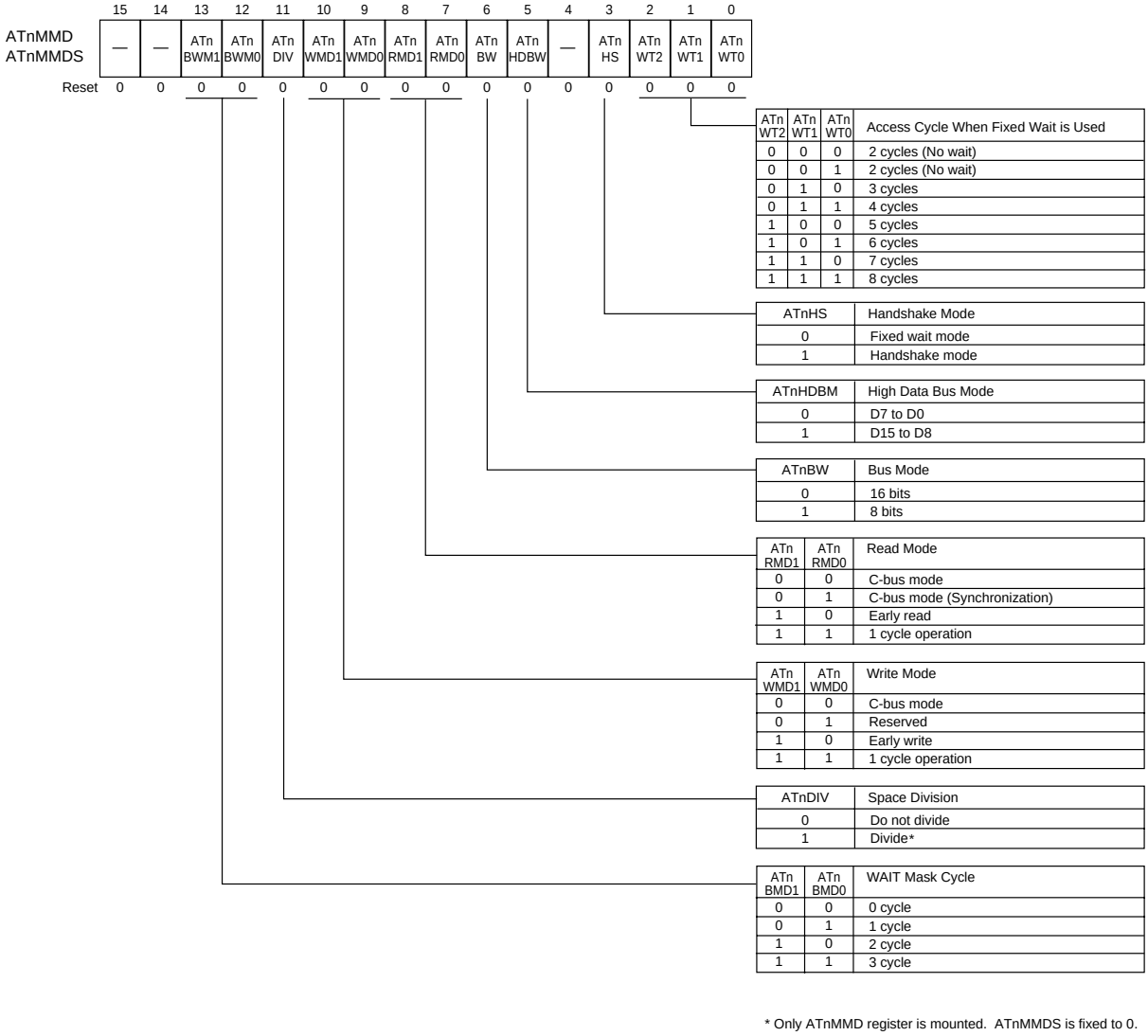

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ATnMMD
ATnMMDS
— — ATn BWM1 ATn BWM0 ATn DIV ATn WMD1 ATn WMD0 ATn RMD1 ATn RMD0 ATn BW ATn HDBW — ATn HS ATn WT2 ATn WT1 ATn WT0
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ATn WT2 | ATn WT1 | ATn WT0 | Access Cycle When Fixed Wait is Used
0 0 0 2 cycles (No wait)
0 0 1 2 cycles (No wait)
0 1 0 3 cycles
0 1 1 4 cycles
1 0 0 5 cycles
1 0 1 6 cycles
1 1 0 7 cycles
1 1 1 8 cycles
ATnHS | Handshake Mode
0 Fixed wait mode
1 Handshake mode
ATnHDBM | High Data Bus Mode
0 D7 to D0
1 D15 to D8
ATnBW | Bus Mode
0 16 bits
1 8 bits
ATn RMD1 | ATn RMD0 | Read Mode
0 0 C-bus mode
0 1 C-bus mode (Synchronization)
1 0 Early read
1 1 1 cycle operation
ATn WMD1 | ATn WMD0 | Write Mode
0 0 C-bus mode
0 1 Reserved
1 0 Early write
1 1 1 cycle operation
ATnDIV | Space Division
0 Do not divide
1 Divide*
ATn BMD1 | ATn BMD0 | WAIT Mask Cycle
0 0 0 cycle
0 1 1 cycle
1 0 2 cycle
1 1 3 cycle
* Only ATnMMD register is mounted. ATnMMDS is fixed to 0.

Figure 9-2-2 ATC Memory Mode Register (Word Access Only, R/W)

[Flag Explanation]

ATnWT[2:0] Select the access cycle when the fixed wait mode is selected. This flag is valid only when the ATnHS is 0 (fixed wait mode). The minimum cycle is 2 cycles. When these flags are ‘000’ and ‘001’, 2 cycles are selected.

ATnHS Specify either fixed wait mode or handshake mode.

ATnHDBM Specify the lower 8 bits or the higher 8 bits when the ATnBW is 1 (8-bit bus width). This flag is not valid when ATnBW is 0 (16-bit bus width).

ATnBW Specify the bus width for applied space.

ATnRMD[1:0] Select the bus mode when the data is read. The following describes the difference between C-bus mode and C-bus mode (Synchronization).

C-bus mode: Assert the NRE signal on the half cycle of the first cycle as C-bus mode of CPU.
C-bus mode (synchronization): Assert the NRE signal at the beginning of the second cycle.
Refer to “9-3-5”
1-cycle mode is an access mode for synchronous memory.
Refer to “9-3-6”

ATnWMD[1:0] Select the bus mode when the data is written. Refer to “9-3-5”, “9-3-6”

ATnDIV Divide the 4-MB space into 2 areas.

ATnBWM[1:0] Select the cycle for wait detect start in the handshake mode. Refer to “9-3-7”

Select C-bus mode synchronization (ATnRMD[1:0] = ‘01’) at the read operation in the handshake mode (ATnHS = ‘1’). Other read operation is desable.

Select early write (ATnWMD[1:0] = ‘10’) for write access to the internal RAM in single address mode. Other write operation is disable.

■ ATC Interrupt Register (ATIRQ)

The ATIRQ register controls interrupts. The NINATC signal sends an interrupt from ATC to CPU. The NINATC signal is generated by the logical OR of AT0ID to AT3ID.

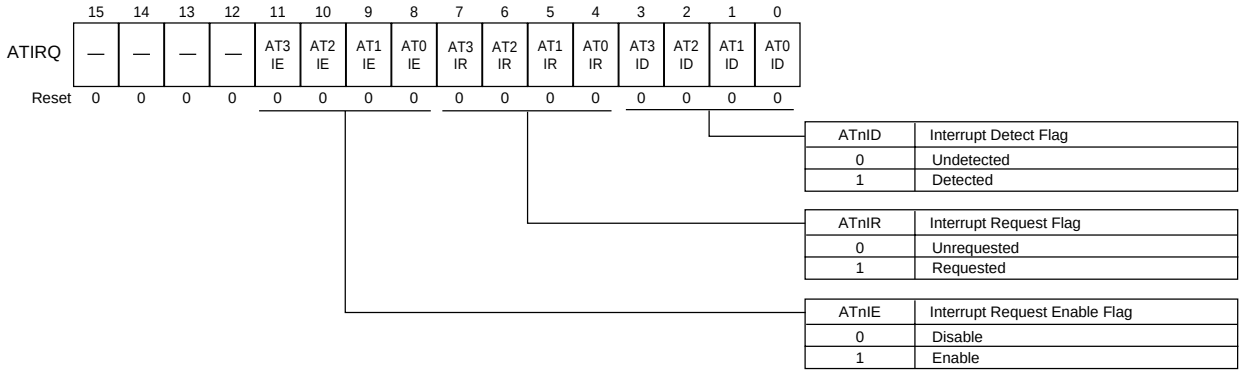

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ATIRQ — — — — AT3 IE AT2 IE AT1 IE AT0 IE AT3 IR AT2 IR AT1 IR AT0 IR AT3 ID AT2 ID AT1 ID AT0 ID
Reset 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ATnID Interrupt Detect Flag
0 Undetected
1 Detected
ATnIR Interrupt Request Flag
0 Unrequested
1 Requested
ATnIE Interrupt Request Enable Flag
0 Disable
1 Enable

Figure 9-2-3 ATC Interrupt Register (Byte/Word Access, R/W)

[Flag Explanation]

ATnIE — Enable or disable an interrupt request. This flag can be read or written.

ATnIR — Request an ATC transfer end interrupt. This flag can be read or written. The ATnIR flag is written only when the ATnIR flag does not change after reading. This prevents the interrupt loss when multiplex interrupts occur.

ATnID — Detect an interrupt. The ATnIE is ANDed with the ATnIR and the result is stored in the ATnID flag. The CPU cannot write this flag.

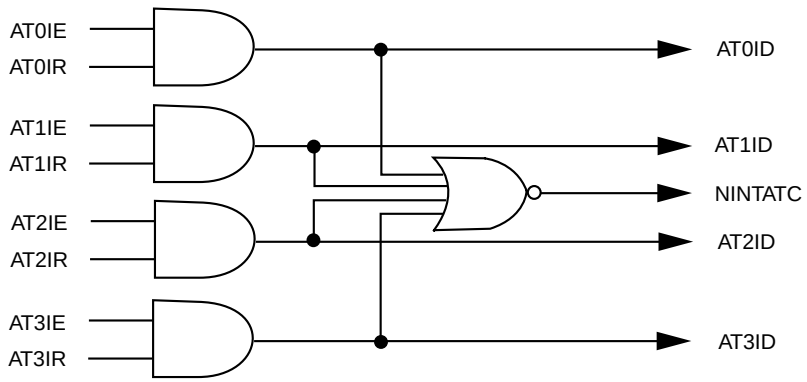


Figure 9-2-4 ATC Interrupt Signal

■ ATC Transfer Word Count Register (ATnCNT)

The ATnCNT register has different functions depending on read operation and write operation. When the data is written to the ATnCNT register, the ATnCNT register is a preset register for transfer word count setup. Set the value of the number to be transferred subtracted by 1 to this register. For example, set “00FF” to this register to transfer 256 words. The maximum of 128 KB transfer is possible when the ATnTBW flag of the ATnCTR register is set to 1 (to select word as transfer unit). When the data is read from the ATnCNT register, the ATnCNT register shows the number of remain transfer during the transfer and the preset value (the last setup value) at transfer end. Therefore, the number of remain transfer is read during 1-word transfer though the preset value is read during burst transfer.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATnCNT	ATn CNT15	ATn CNT14	ATn CNT13	ATn CNT12	ATn CNT11	ATn CNT10	ATn CNT9	ATn CNT8	ATn CNT7	ATn CNT6	ATn CNT5	ATn CNT4	ATn CNT3	ATn CNT2	ATn CNT1	ATn CNT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-5 ATC Transfer Word Count Register (Word Access Only, R/W)

■ ATC Source Address Register (ATnSRC)

The ATnSRC register specifies source address for the ATC transfer. When the ATnTBW flag of the ATnCTR register is 1 to select word as the transfer unit, the LSB of this register is set to ‘0’ (even address) . If the LSB of this register is set to ‘1’, an error may occur. When the data is read, the ATnSRC register shows the next source address. When the source address is increment, the ATnSRC register specifies the latest source address. Set the ATnSRC register again when the transfer is completed. **The 24 bits of ATnSRC23 to ATnSRC0 must be written at once with the MOVX instruction to update this register. This register cannot be updated if only 16 bits of ATnSRC15 to ATnSRC0 are written with the MOV instruction.** This register is updated when ATnSRC23 to ATnSRC16 are written . When this register is updated using with the MOV instruction, update the lower 16 bits first and then the upper 8 bits. This prevents from transferring the data to the wrong address while this register is updated with the MOV instruction.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATnSRC0	ATn SRC15	ATn SRC14	ATn SRC13	ATn SRC12	ATn SRC11	ATn SRC10	ATn SRC9	ATn SRC8	ATn SRC7	ATn SRC6	ATn SRC5	ATn SRC4	ATn SRC3	ATn SRC2	ATn SRC1	ATn SRC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATnSRC1	-	-	-	-	-	-	-	-	ATn SRC23	ATn SRC22	ATn SRC21	ATn SRC20	ATn SRC19	ATn SRC18	ATn SRC17	ATn SRC16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-6 ATC Source Address Register (Byte/Word Access, R/W)

■ ATC Destination Address Register (ATnDST)

The ATnDST register specifies destination address for the ATC transfer. When the ATnTBW flag of the ATnCTR register is 1 to select word as the transfer unit, the LSB of this register is set to '0' (even address) . If the LSB of this register is set to '1', an error may occur. When the data is read, the ATnDST register shows the next destination address. When the destination address is increment, the ATnDST register specifies the latest destination address. Set the ATnDST register again when the transfer is completed. **The 24 bits of ATnDST23 to ATnDST0 must be written at once with the MOVX instruction to update this register. This register cannot be updated if only 16 bits of ATnDST15 to ATnDST0 are written with the MOV instruction.** This register is updated when ATnDST23 to ATnDST16 are written . When this register is updated using with the MOV instruction, update the lower 16 bits first and then the upper 8 bits. This prevents from transferring the data to the wrong address while this register is updated with the MOV instruction.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATnDST0	ATn DST15	ATn DST14	ATn DST13	ATn DST12	ATn DST11	ATn DST10	ATn DST9	ATn DST8	ATn DST7	ATn DST6	ATn DST5	ATn DST4	ATn DST3	ATn DST2	ATn DST1	ATn DST0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATnDST1	-	-	-	-	-	-	-	-	ATn DST23	ATn DST22	ATn DST21	ATn DST20	ATn DST19	ATn DST18	ATn DST17	ATn DST16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-7 ATC Destination Address Register (Byte/Word Access, R/W)

■ ATC Transfer Channel Priority Setup Register (ATPRI)

The ATPRI register determines the priority order among transfer channels. When transfer requests occur from more than two devices, ATC asserts the bus request to the CPU. Then the CPU sends an enable signal back to ATC, and the ATPRI register determines the priority order of transfer channels. ATC determines the priority order each time the transfer ends. ATC negates the bus request to the CPU when a sequence of all transfers ends.

The following is the priority order.
{ATnPRI1, ATnPRI0}:{0,0}>{0,1}>{1,0}>{1,1}

When the same priority orders are set, the channel with the smaller number has the priority.
For example, both {AT0PRI0, AT0PRI1} and {AT1PRI0, AT1PRI1} are set to {1,0},
{AT0PRI0, AT0PRI1}>{AT1PRI0, AT1PRI1}

This Priority rule is applied regardless of software activation or hardware activation.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATPRI	-	-	-	-	-	-	-	-	AT3 PRI1	AT3 PRI0	AT2 PRI1	AT2 PRI0	AT1 PRI1	AT1 PRI0	AT0 PRI1	AT0 PRI0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-8 ATC Transfer Channel Priority Setup Register (Byte/Word Access, R/W)

■ ATC Input/Output Channel Select Register (ATSEL)

The ATSEL register selects an input/output channel. The ATSEL register switches input port or output port. Therefore, do not change this register during request or transfer.

Select ATRQnA, ATACKnA when ATnSEL is 0.
Select ATRQnB, ATACKnB when ATnSEL is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATSEL	-	-	-	-	-	-	-	-	-	-	-	-	AT3 SEL	AT2 SEL	AT1 SEL	AT0 SEL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-9 ATC Input/Output Channel Select Register (Byte/Word Access, R/W)

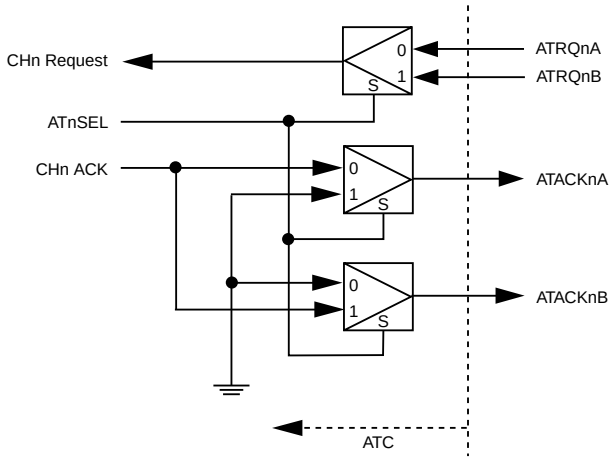


Figure 9-2-10 ATC Input/Output Channel Select Diagram

The ATACKnA signal or ATACKnB signal corresponded to each endpoint is output when the data is transferred between USB-FIFO and memory. The ATACKnA signal and ATACKnB signal become FIFO select signals in USB-FIFO single address mode transfer. Select each endpoint based on the following table.

Table 9-2-3 Endpoint Select List in Single Address Mode

ATACK0A ATACK0B	Endpoint 1 Endpoint 2
ATACK1A ATACK1B	Endpoint 3 Endpoint 4
ATACK2A ATACK2B	Endpoint 5 Endpoint 6
ATACK3A ATACK3B	Endpoint 7 Endpoint 8

■ ATC Hardware Activation Factor Select Register (ATRQSEL)

The ATRQSEL register selects a hardware activation factor to start ATC. The ATRQSEL register and the ATSEL register determine the hardware activation factor. Do not change the ATRQSEL register during request or transfer because it only selects a hardware activation factor.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATRQSEL	-	-	-	-	-	-	-	-	ATRQ 7	ATRQ 6	ATRQ 5	ATRQ 4	ATRQ 3	ATRQ 2	ATRQ 1	ATRQ 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-2-11 ATC Hardware Activation Factor Select Register (Byte/Word Access, R/W)

Table 9-2-3 shows each bit meaning.

Table 9-2-4 ATRQSEL Register Setup

	ATRQ Signal	Value	Trigger Factors
ATRQ[0]	ATRQ0A	0	Serial 0 Reception End
		1	Timer 4 Underflow
ATRQ[1]	ATRQ0B	0	Serial 0 Transmission End
		1	Timer 5 Underflow
ATRQ[2]	ATRQ1A	0	Serial 1 Reception End
		1	Timer 6 Underflow
ATRQ[3]	ATRQ1B	0	Serial 1 Transmission End
		1	Timer 7 Underflow
ATRQ[4]	ATRQ2A	0	USBSOF Interrupt
		1	Serial 2 Reception End
ATRQ[5]	ATRQ2B	0	Timer 2 Underflow
		1	Serial 2 Transmission End
ATRQ[6]	ATRQ3A	0	A/D Conversion End
		1	Serial 3 Reception End
ATRQ[7]	ATRQ3B	0	Timer 3 Underflow
		1	Serial 3 Transmission End

9-3 ATC Operation

9-3-1 Software Activation/Hardware Activation

The ATnIQ flag of the ATnCTR register selects software or hardware to activate ATC.

Setting the ATnEX flag of the ATnCTR register to 1 activates ATC by software. The data is transferred together in the burst transfer mode. On the other hand, the ATRQnA signal or ATRQnB signal activates ATC by hardware and the data is transferred in the burst transfer mode or 1-word transfer mode. The ATnTMD flag selects the burst transfer mode or 1-word transfer mode.

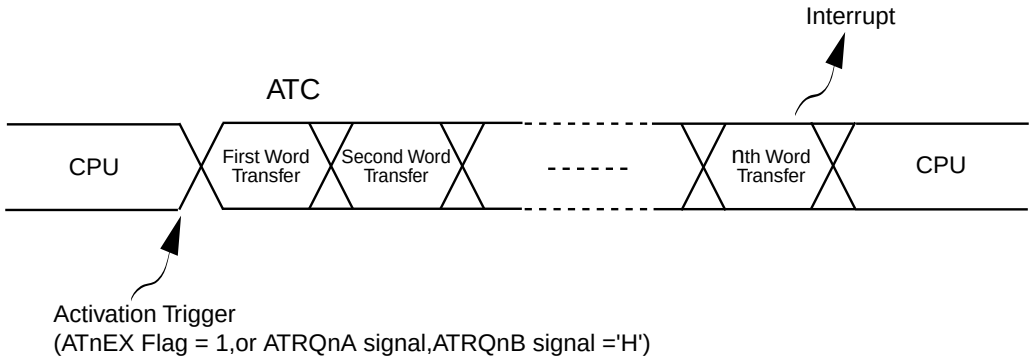


Figure 9-3-1 ATC Operation

9-3-2 Transfer Sequence

When ATC is activated by a activation factor, ATC sends a bus request to the CPU through a NBREQ signal. When the CPU accepts a bus request, the CPU informs ATC that bus is open through a NBRACK signal. After that, ATC starts transferring the data. ATC returns the bus when ATC completes the necessary number of transfers (one transfer when 1 word transfer is set). When all data has transferred, ATC sends an interrupt to the CPU. ATC has two transfer modes of the burst transfer mode and 1-word transfer mode when ATC is activated by hardware. In the burst transfer mode, all number of the data set in the ATnCNT register is transferred each time ATC is activated by a hardware activation factor (ATRQn). In 1-word transfer mode, the data is transferred once each time ATC is activated by a hardware activation factor. ATC sends an interrupt when all transfers have completed.

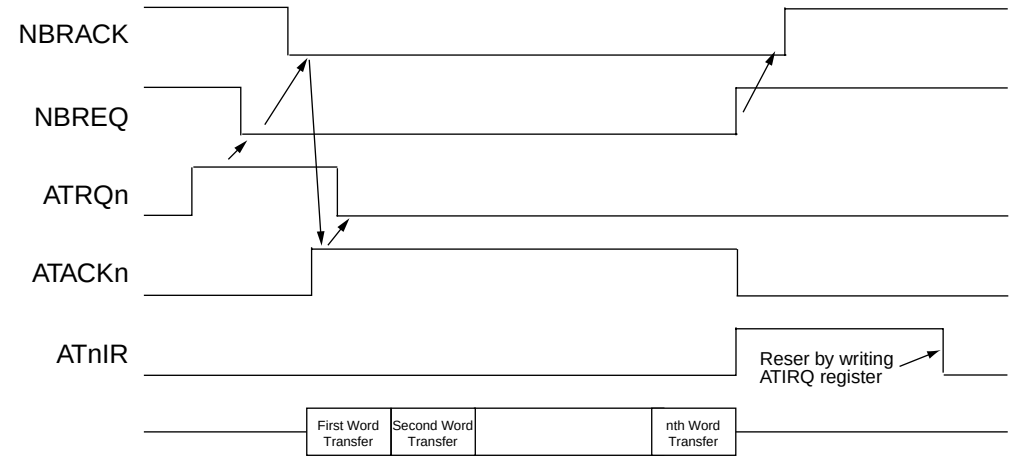


Figure 9-3-2 Burst Transfer Mode Sequence

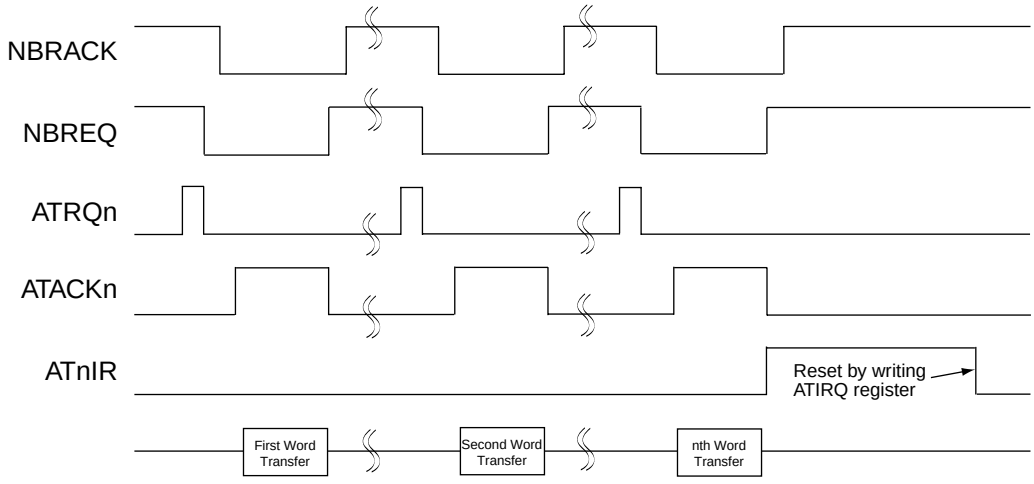


Figure 9-3-3 1 Word Transfer Mode Sequence

When ATC transfers the data through the same channel, the ATRQn signal must be asserted again while the ATACKn signal is negated. ATC accepts a bus request when an ATRQn signal is negated and an ATRQn signal is asserted.

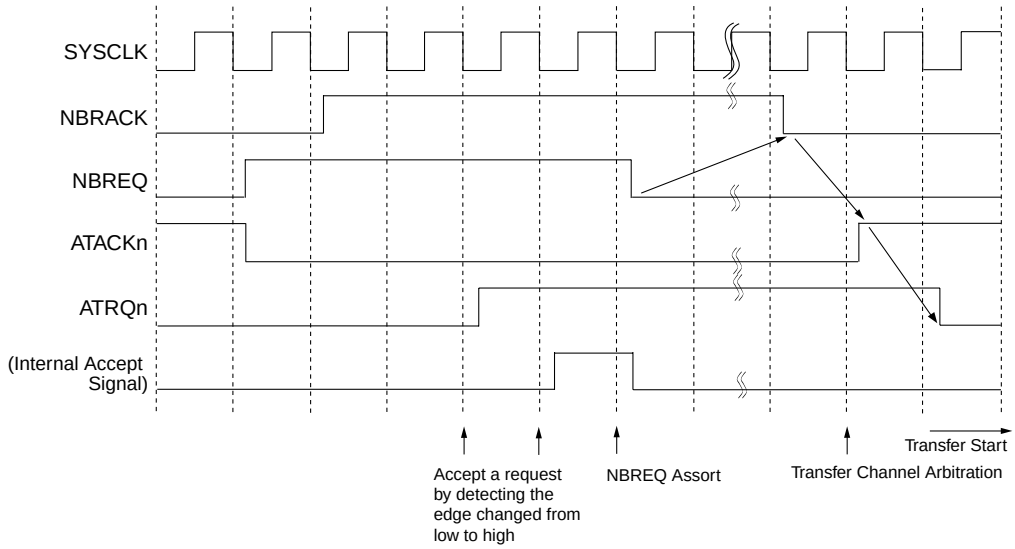


Figure 9-3-4 ATC Transfer Sequence for Single Channel

Setting the associated IR flag of the ATIRQ register and resetting the transfer enable flag (ATnEN) of the ATnCTR register end the data transfer activated by a hardware activation factor. This prevents from starting the next data transfer automatically before the next data transfer is prepared during the interrupt service routine. Because ATC accepts the transfer request while the ATnEN flag is reset, the following processes are executed when the ATnEN flag is set.

1. ATC sends a bus request to the CPU.
2. ATC executes the transfer channel arbitration.

When transfer requests occur consecutively in a short time, the transfer request occurs before setting the ATnEN flag again during the interrupt service routine. In this case, setting the ATnENX flag (to invalid the ATnEN flag) allows ATC to accept the transfer request regardless of the ATnEN value. Careful attention is needed to manage the data because the address register is increment or decrement automatically by starting the transfer.

In the transfer by hardware activation, when the transfer in set times ends, ATnEN flag is cleared. However, if the same activation factor is generated after the transfer ends, ATC is activated as soon as ATnEN flag is enabled next time. The ATC accepts a transfer request by asserting the ATRQn flag regardless the ATnEN flag setup, therefore ATC can cancel the accepted transfer request before this by software activation. When the ATnIQ flag is set to software activation, ATC cancels all transfer requests and does not accept a new request even though the ATRQn signal is asserted.

9-3-3 Transfer Sequence for Multiple Channels

When ATC accepts transfer requests from more than two channels at the same time, ATC continues the data transfer for these channels without returning the bus authority to the CPU.

The following shows the conditions when ATC accepts transfer requests.

1. The ATACKn for the same channel is low (The previous transfer must be completed.)
2. The ATnCTR register selects hardware activation.

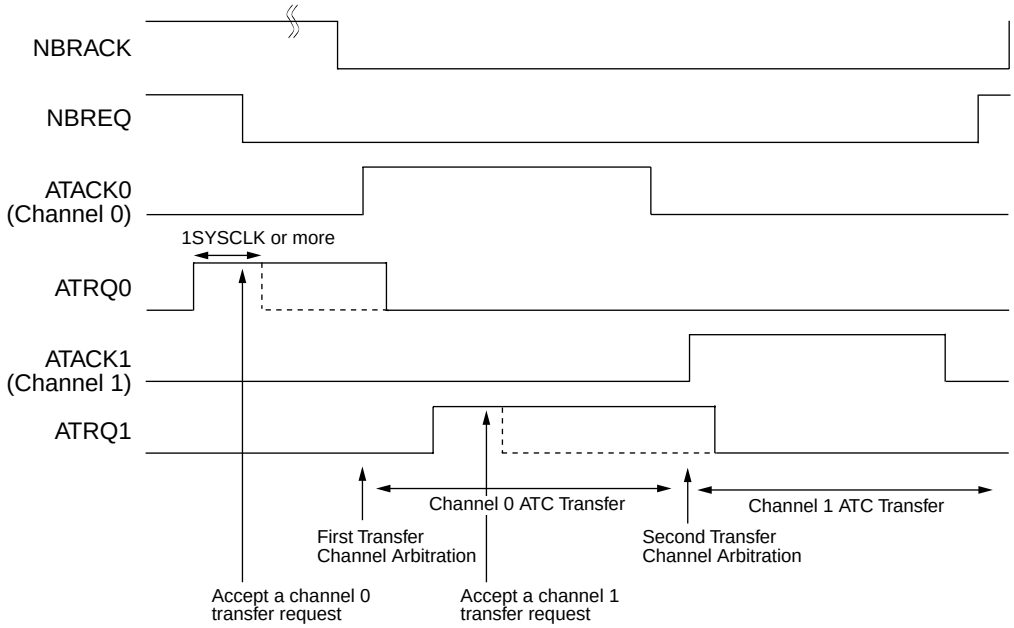


Figure 9-3-5 ATC Transfer Sequence for Multiple Channels

9-3-4 Single Address Mode/Dual Address Mode

In the single address mode, ATC executes read and write operations in one bus transaction. In the dual address mode, ATC executes read and write operations in two bus transactions. The single address mode is used for the high-speed transfer between USB-FIFO and the internal RAM.

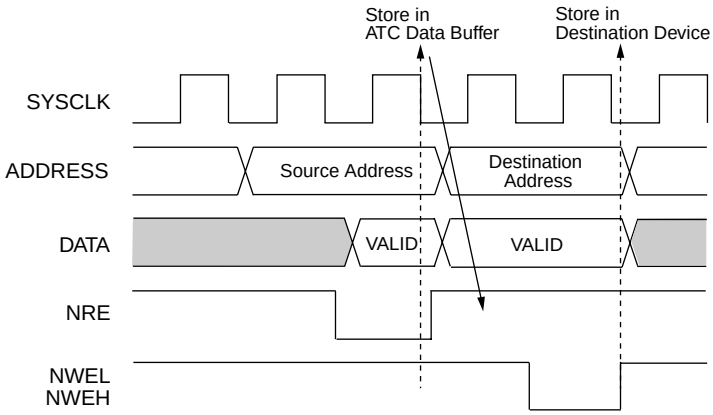


Figure 9-3-6 Dual Address Mode

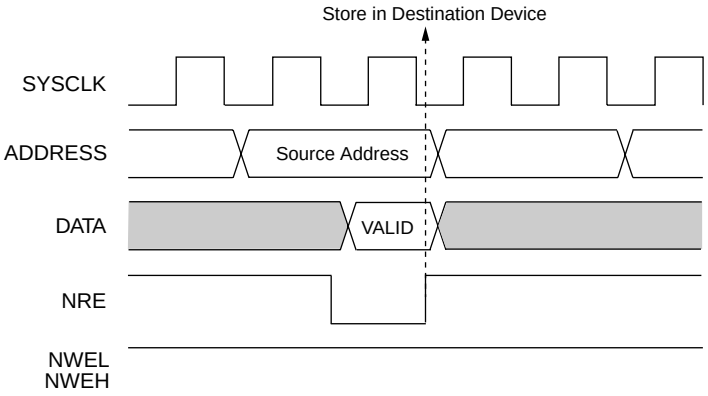


Figure 9-3-7 Single Address Mode (Source Address Specification)

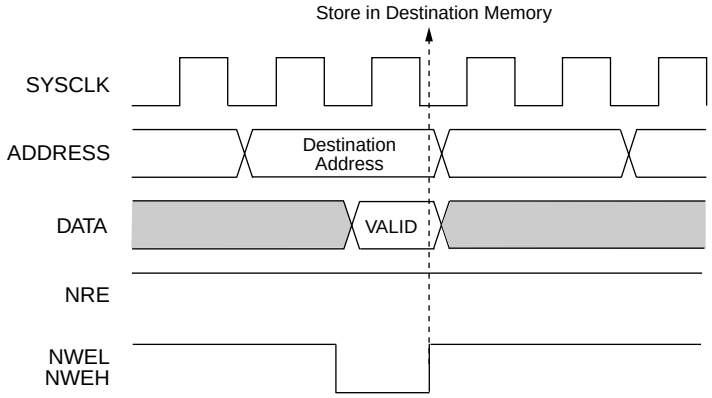


Figure 9-3-8 Single Address Mode (Destination Address Specification)

In the single address mode, the data is transferred between the internal RAM and USB-FIFO which has the special interface corresponded to the single address mode in 1 bus transaction. When the internal RAM is used as the source, the single address mode with source address specification is selected. On the other hand, when the internal RAM is used as the destination, the single address mode with destination address specification is selected.

9-3-5 C-bus Mode, Early Read Mode and Early Write Mode

Two C-bus modes are available during read operation; the mode of requiring 1.5 cycles for the read signal assert time and the mode of requiring 1 cycle for the read signal assert time.

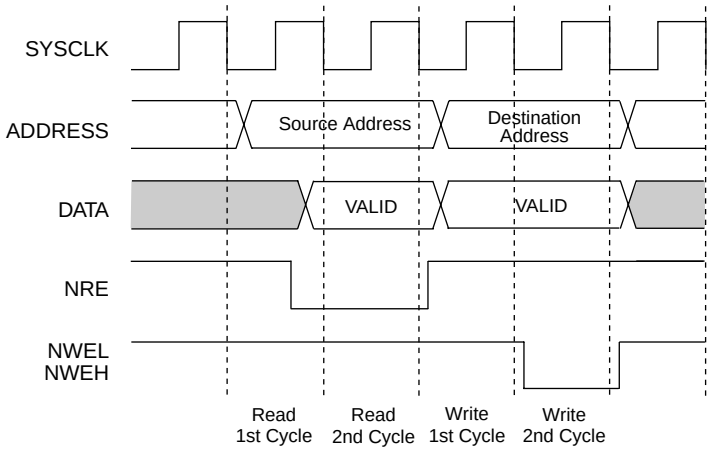


Figure 9-3-9 C-bus Mode Base Cycle (No Wait, 2 Cycles)

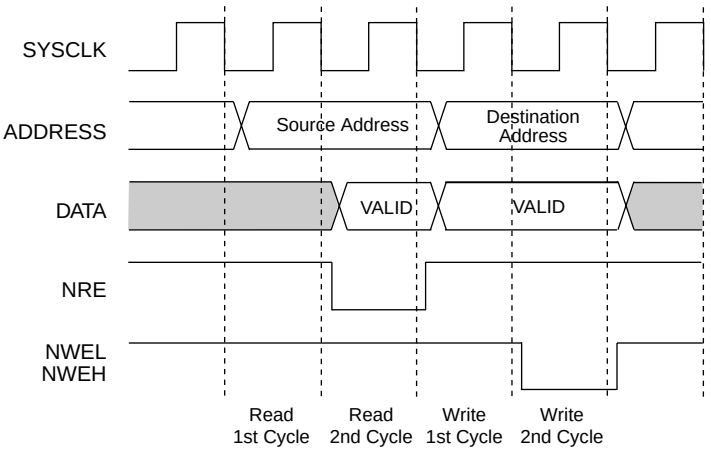


Figure 9-3-10 C-bus Mode Base Cycle (Read Synchronization)

The read signal and the write signal are asserted 0.5 cycle forward when the early read mode and the early write mode are selected.

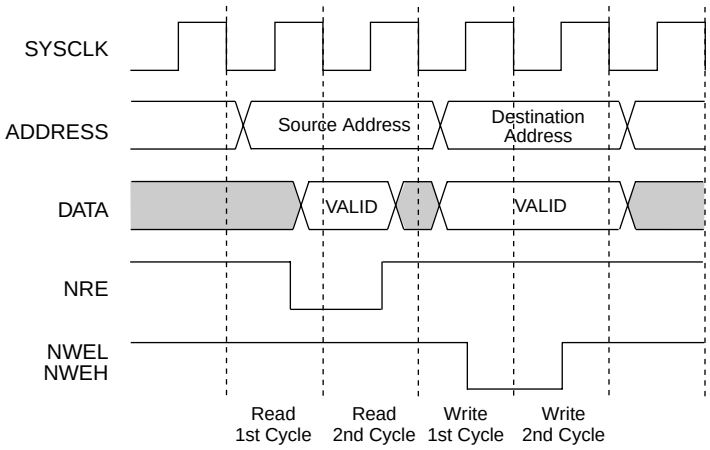


Figure 9-3-11 Early Read/Early Write (No Wait, 2 Cycles)

9-3-6 1 Cycle Operation Mode

The c-bus mode described in 9-3-5 section has the minimum of 2 cycles. 1 access cycle is possible only to the synchronous memory (ROM, RAM) by setting the ATnRMD flag and the ATnWMD flag of the ATnMMD register and the ATnMMDS register.

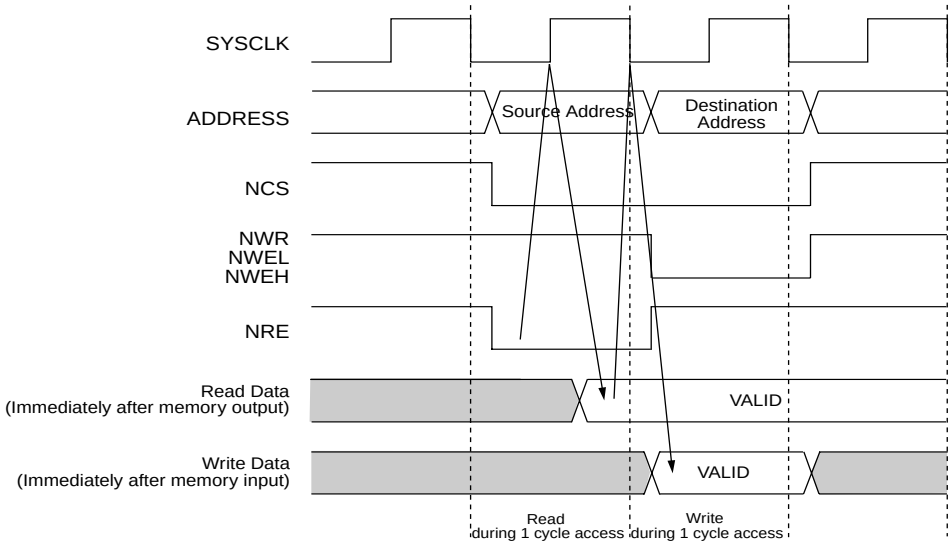


Figure 9-3-12 1 Cycle Operation Mode

In the area of built-in RAM and I/O control register, it is disable to access by the 1 cycle operation mode.

9-3-7 Wait Control

ATC can extend the bus cycle by controlling waits. The following two modes are used to control waits; the fixed wait mode set by software and the handshake mode set by hardware.

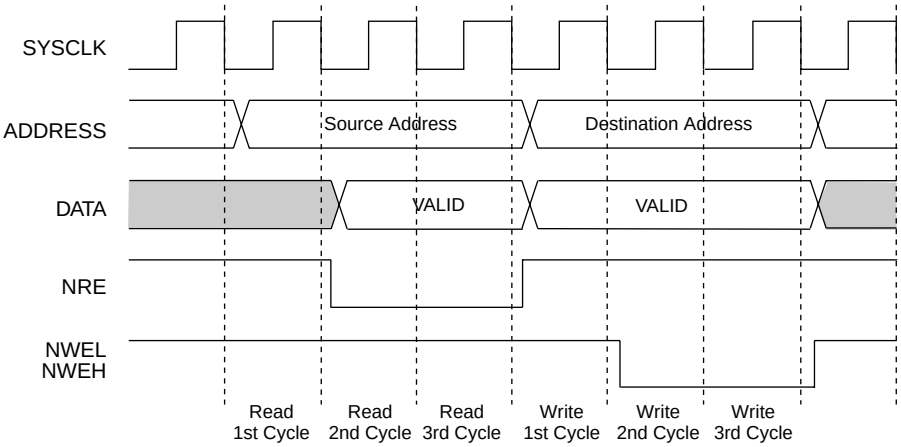


Figure 9-3-13 C-bus Mode (1 Wait, 3 Cycles)

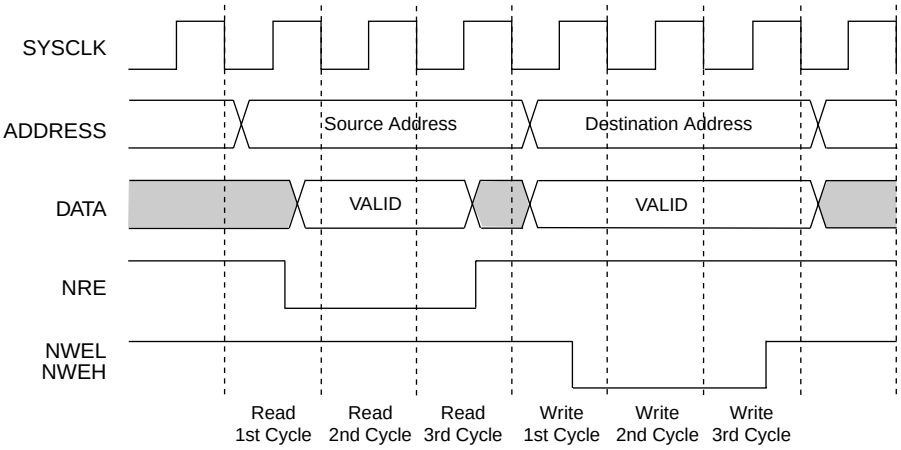


Figure 9-3-14 Early Read/Early Write Mode (1 Wait, 3 Cycles)

ATC can control waits by setting the associated pin in the handshake mode. The following figure is the example of setting the bus wait mask cycle to 0 cycle. When the ATnBMD[1:0] flags of the ATnMMD register and the ATnMMDS register are set to '00' (the wait mask cycle is 0 cycle), detecting the WAIT signal starts on the rising edge of the second cycle of SYSCLK. In this example, the bus cycle is 3 cycles because the wait is inserted during the second cycle and the wait is cleared during the third cycle.

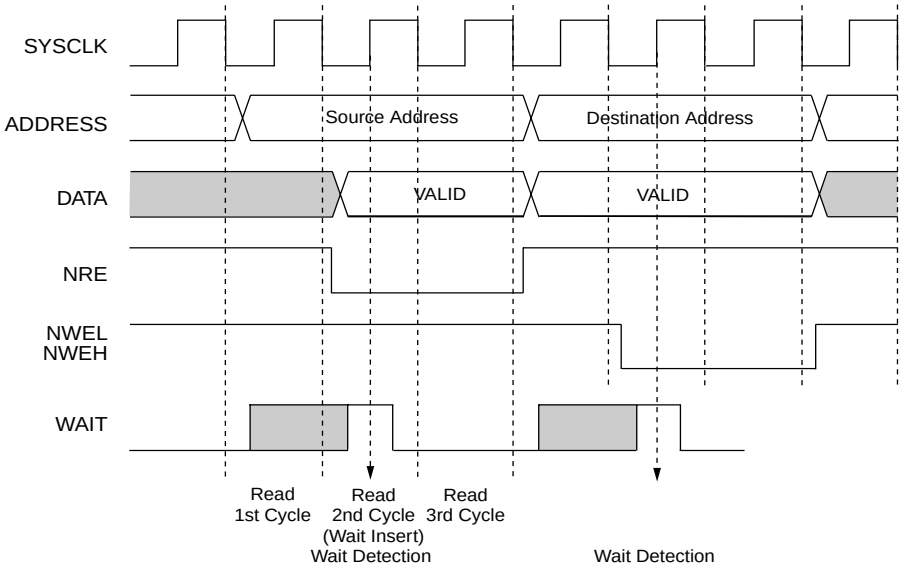


Figure 9-3-15 1 Wait Insert by Wait Signal in Handshake Mode
WAIT Mask Cycle : 0 Cycle

Setting the ATnBMD[1:0] flags (the wait mask cycle) of the ATnMMD register and the ATnMMDS register changes the start cycle for wait detection. As shown below, when the ATnBMD[1:0] flags are set to '01', the wait detection starts on the third cycle.

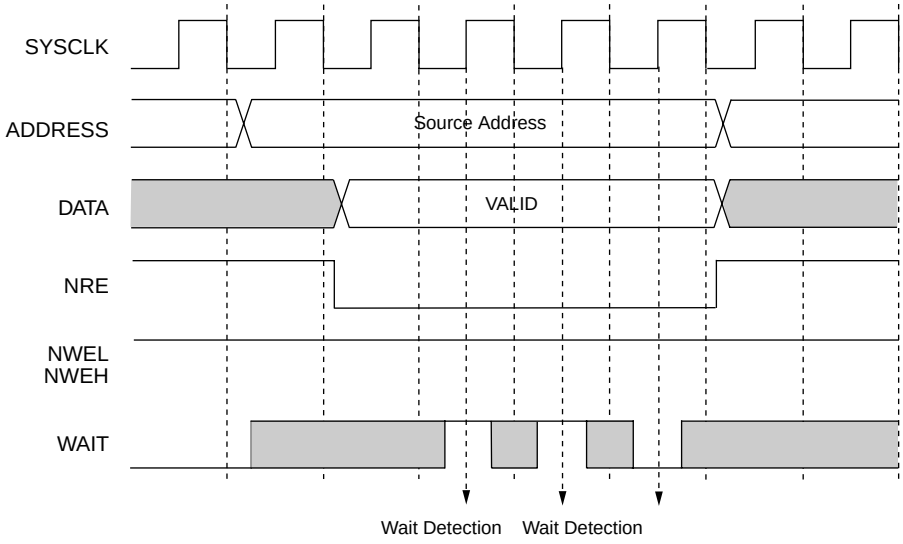


Figure 9-3-16 2 Cycle Insert by Wait Signal in Handshake Mode
WAIT Mask Cycle : 1 Cycle

9-3-8 ATC Transfer

When the source address and the destination address differ and word transfer is selected, the bus cycle occurs twice on one-way transfer.

Example 1
Source Address: 8-bit bus width
Destination Address: 16-bit bus width
Data Transfer Unit: Word

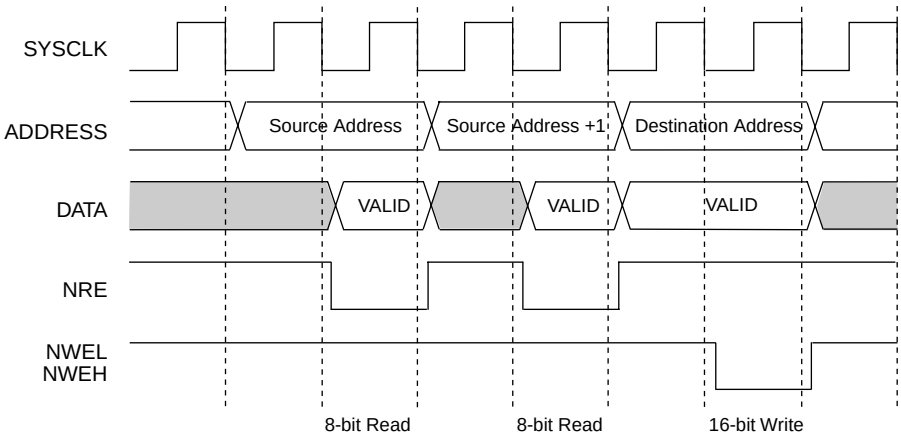


Figure 9-3-17 Transfer Example 1

> Set both source address and destination address to even address when word transfer is selected regardless of 16-bit bus width or 8-bit bus width. Data cannot be transferred normally due to alignment error when those addresses are set to odd addresses.

> The ATnHDBM flag of the ATnMMD register specifies either the upper bits (D15 to D8) or the lower bits (D7 to D0) of bus in 8-bit bus space.

Example 2
Source Address: 16-bit bus width
Destination Address: 8-bit bus width
Data Transfer Unit: Word

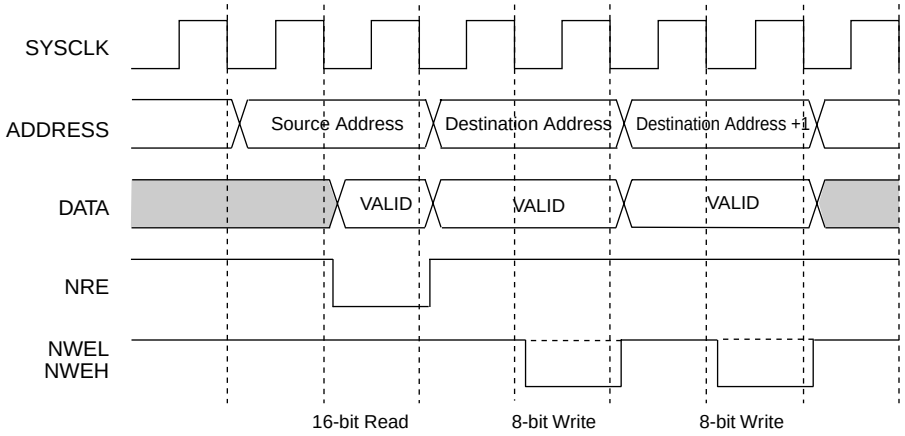


Figure 9-3-18 Transfer Example 2

Set both source address and destination address to even address when word transfer is selected regardless of 16-bit bus width or 8-bit bus width. Data cannot be transferred normally due to alignment error when those addresses are set to odd addresses.

The ATnHDBM flag of the ATnMMD register specifies either upper bits (D15 to D8) or lower bits (D7 to D0) of bus in 8-bit bus space.

Example 3
Source Address: 8-bit bus width
Destination Address: 16-bit bus width
Data Transfer Unit: Byte

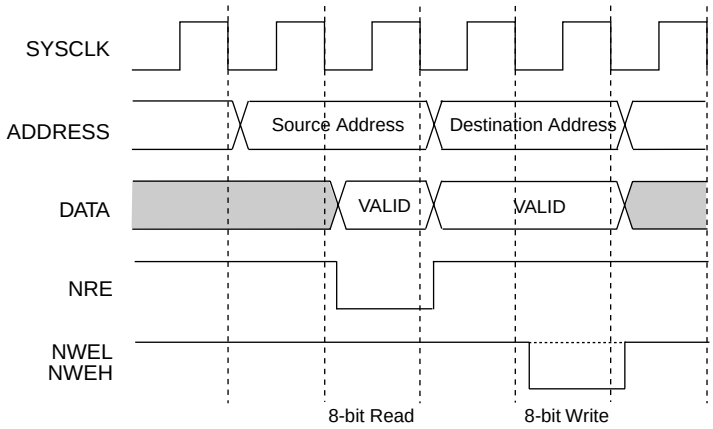
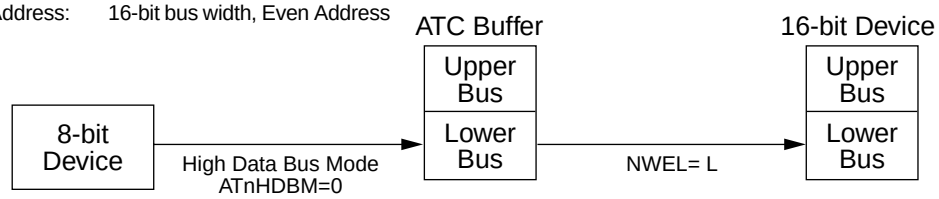


Figure 9-3-19 Transfer Example 3

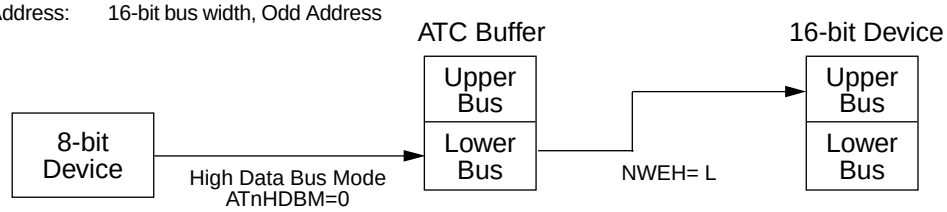
In this example, set source address and destination address to either even address or odd address due to no alignment restrictions because byte transfer is selected.

In example 3, the actual data transfer has the following four choices. As example 3-2 and example 3-3, the bus switches the upper bits or the lower bits in ATC.

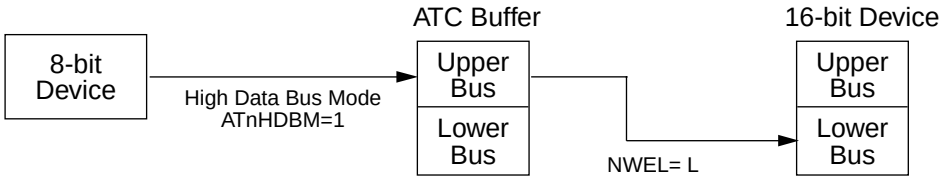
Example 3-1
Source Address: 8-bit bus width, ATnHDBM=0
Destination Address: 16-bit bus width, Even Address



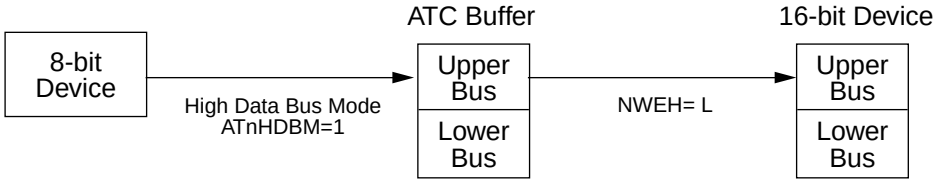
Example 3-2
Source Address: 8-bit bus width, ATnHDBM=0
Destination Address: 16-bit bus width, Odd Address



Example 3-3
Source Address: 8-bit bus width, ATnHDBM=1
Destination Address: 16-bit bus width, Even Address



Example 3-4
Source Address: 8-bit bus width, ATnHDBM=1
Destination Address: 16-bit bus width, Odd Address



Example 4
Source Address: 16-bit bus width
Destination Address: 8-bit bus width
Data Transfer Unit: Byte

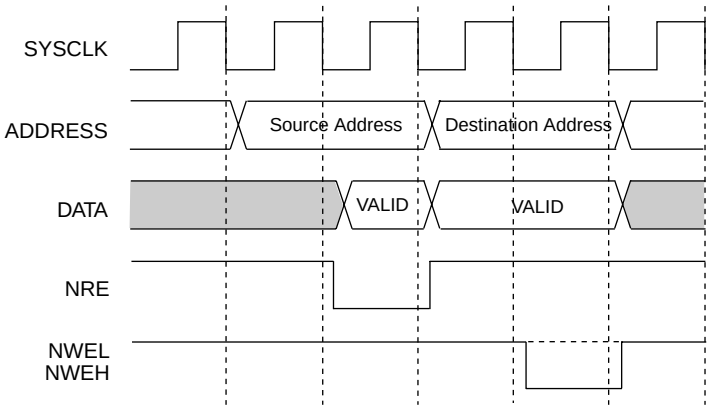
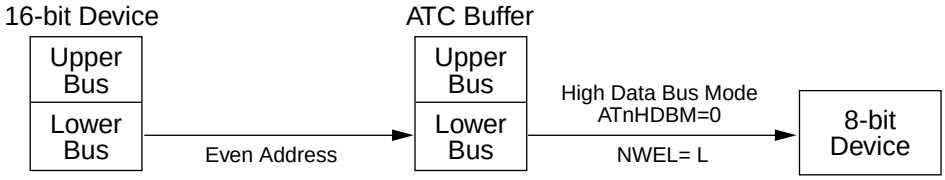


Figure 9-3-20 Transfer Example 4

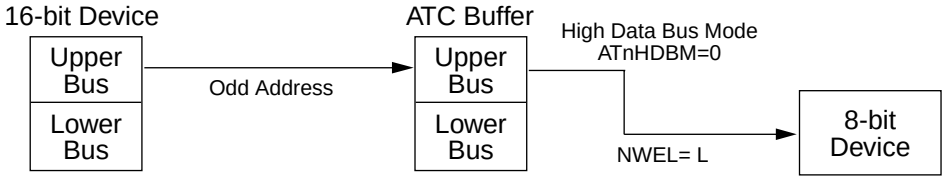
> In this example, set source address and destination address to either even address or odd address due to no alignment restrictions because byte transfer is selected.

The actual data transfer has the following four choices as example 3.

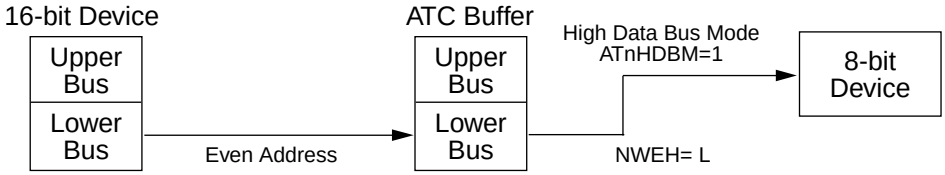
Example 4-1
Source Address: 16-bit bus width, Even Address
Destination Address: 8-bit bus width, ATnHDBM=0



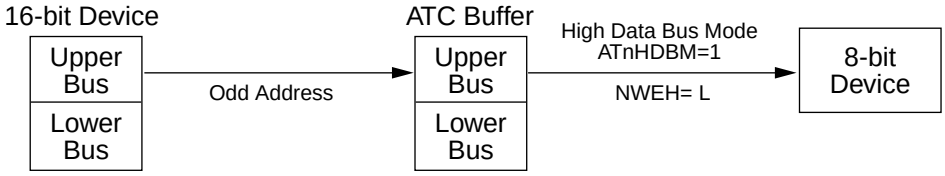
Example 4-2
Source Address: 16-bit bus width, Odd Address
Destination Address: 8-bit bus width, ATnHDBM=0



Example 4-3
Source Address: 16-bit bus width, Even Address
Destination Address: 8-bit bus width, ATnHDBM=1



Example 4-4
Source Address: 16-bit bus width, Odd Address
Destination Address: 8-bit bus width, ATnHDBM=1



9-3-9 ATC Transfer in Single Address Mode

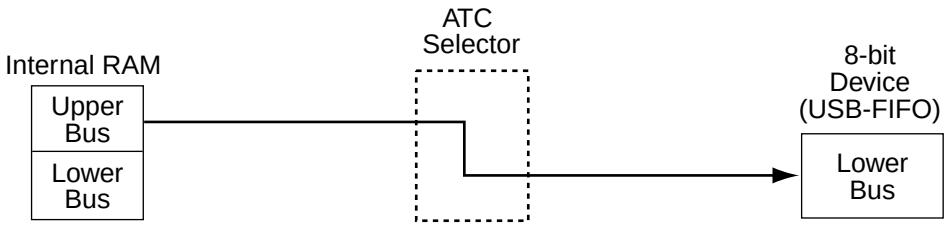
The following two points are restricted when single address mode is selected.

1. The read or write cycle must be completed during one transaction.
2. The bus mode of the source and the bus mode of the destination must be the same.

In single address mode, either source address or destination address is specified depending on which side is the internal RAM set. Also the bus mode operation is determined depending on the specified address side.

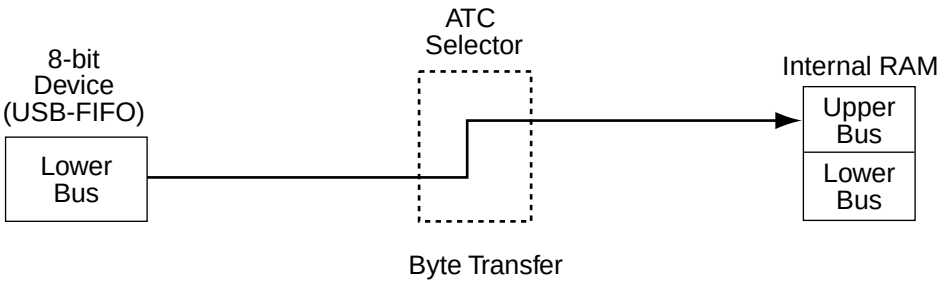
The following shows the data flow when single address mode is selected.
In example 5-1, the data is transferred from the internal RAM to 8-bit device (USB-FIFO) in single address mode.

Example 5-1
Source Address: Internal RAM, Odd Address
Destination Address: 8-bit bus width, ATnHDBM=0
Transfer Unit: Byte



Example 5-2
Source Address: 8-bit bus width, ATnHDBM=0
Destination Address: Internal RAM, Odd Address
Transfer Unit: Byte

In example 5-2, the data is transferred from 8-bit device (USB-FIFO) to the internal RAM in single address mode.



9-3-10 External Bus Request Arbitration

ATC arbitrates between an external bus request and an ATC activation request, requests a bus release to the CPU and informs the bus acquisition to the external device.

The following figure shows the bus competition. ATC cannot asserts a bus release signal (NBRACK) until the transfer is completed even an external bus request is sent to ATC because ATC has transferred the data through ATRQ0. When ATC completes transferring the data through ATRQ0, ATC arbitrates between requests again and moves the bus authority to the external bus master. Even though an ATC transfer request for ATRQ1 occurs, the external bus master has a higher priority so that ATC always pass the bus authority to the external bus master. The external bus master negates a bus request (NBREQ pin input) when the external bus master completes the bus usage. Then ATC arbitrates between requests again and starts the next transfer. When a single bus request occurs, 2 cycles are required from NBREQ acceptance until NBRACK assert. (1 cycle is used to accept a request and another 1 cycle is used to send a request to the CPU.)

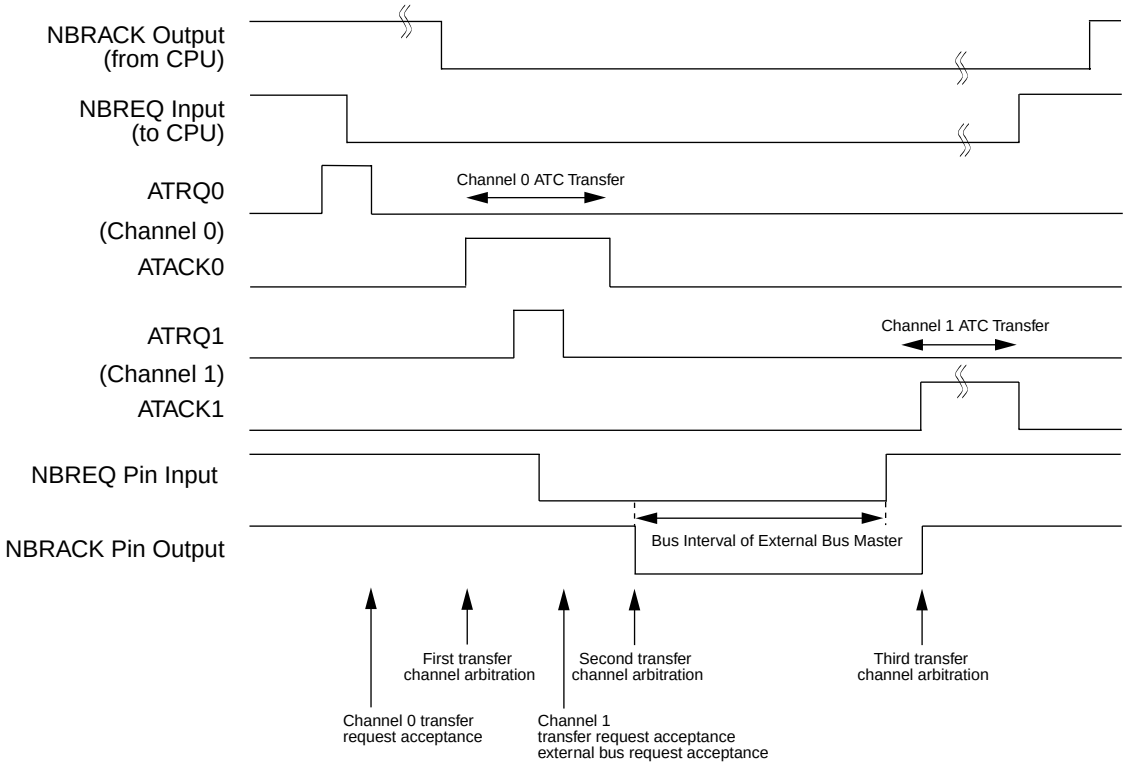


Figure 9-3-21 External Bus Request Arbitration

9-3-11 Transfer Suspend Using Nonmaskable Interrupt (NMI)

ATC suspends the transfer and opens the bus to the CPU when a nonmaskable pin interrupt (NMI) occurs during burst transfer. Clear the transfer start and busy flag (ATnEX) of the channel to be suspended to 0 and set the transfer suspend flag (ATnST) to 1. Set all transfer enable flags (ATnEN) and ATnEN invalid flags to 0. To start the transfer again after interrupt service routine, ATC is activated by software (ATnEX=1) or by hardware (ATnEN=1) because a transfer request is still remained. This clears the transfer suspend flag and starts the transfer. The number of transfer, the source address and the destination address before the transfer suspend are held. When ATC accepts transfer requests from more than two channels at transfer suspend, ATC starts the transfer by setting ATnEN to 1 because ATC holds transfer requests of other channels than the channel which transfer is suspended.

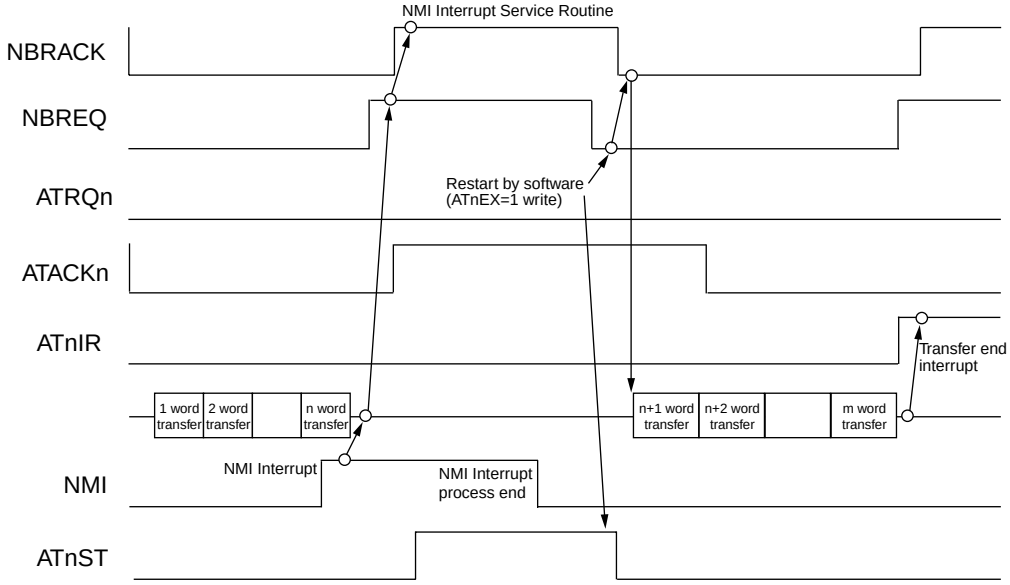


Figure 9-3-22 Transfer Suspend Using NMI

> When a nonmaskable pin interrupt (NMI) occurs during 1-word transfer, clear the transfer start and busy flag (ATnEX) of the channel to be suspended to 0 and set the transfer suspend flag (ATnST) to 1. Set all transfer enable flags (ATnEN) and ATnEN invalid flags to 0. The number of transfer, the source address and the destination address before the transfer suspend are held.

> When a NMI interrupt occurs while ATC stops transferring the data, set the transfer enable flags (ATnEN) and ATnEN invalid flags of all transfer channels to 0.

9-4 ATC Transfer Setup Example

9-4-1 DMA Transfer by Software Activation

ATC transfers the 1 KB data on the 16-bit bus space on the address x'200000' to the 8-bit bus space on the address x'600000' in word units.

■ Setup Procedures

1. Set the ATC memory mode register (ATnMMD, ATnMMDS).

2. Set the ATC transfer word count register (ATnCNT).
The 1 KB data is 512 words in word transfer. Therefore, write x'01FF' (x'0200'-1) to the ATnCNT register.

3. Set the ATC source address register (ATnSRC).
Write x'200000' to the ATnSRC register.

4. Set the ATC destination address register (ATnDST).
Write x'600000' to the ATnDST register.

5. Set the ATC control register (ATnCTR).
Write x'8824' to the ATnCTR register.

Activation factor:	software activation (ATnIQ=0)
Source increment mode:	increment (ATnSI[1:0]=01)
Destination increment mode:	increment (ATDI[1:0]=01)
Addressing mode:	dual (ATnAMD[1:0]=00)
Transfer unit:	word (ATnTBW=1)
Transfer mode:	either way
Transfer enable:	either way
ATnEN invalid:	either way
Transfer start and busy flag:	transfer start (ATnEX=1)

9-4-2 DMA Transfer by Hardware Activation

ATC transfers the data on the 16-bit bus space on the address x'200000' to the fixed I/O address (32 byte FIFO for serial transmission) of the 8-bit bus space on the address x'600000'.

The trigger level for serial interface is 16 bytes. ATC activates by hardware when FIFO becomes less than 16 bytes. Select the burst transfer mode. Set the next transfer again during the interrupt service routine. PIO instead of DMA transfers the first data.

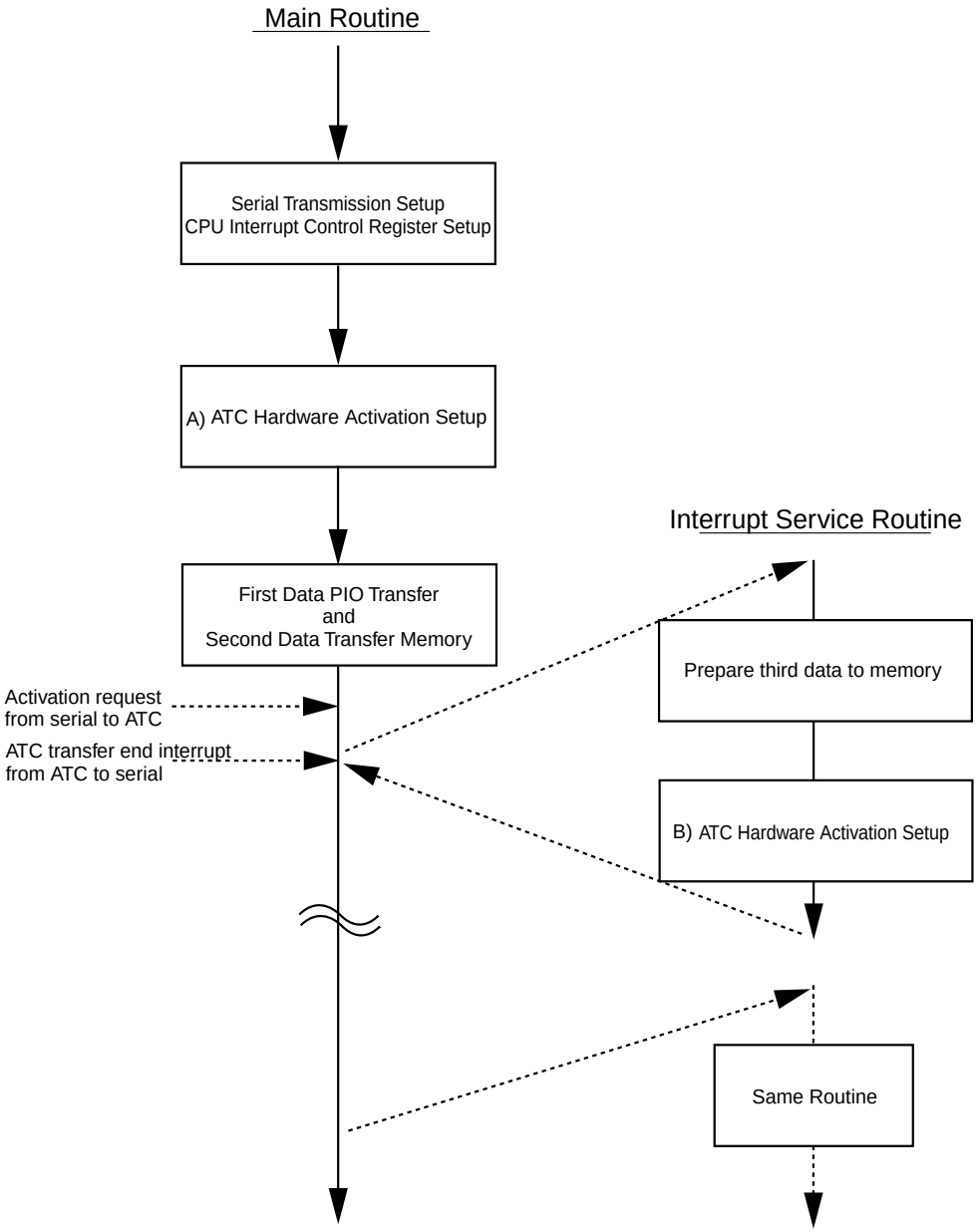


Figure 9-4-1 Process Flow

The procedures at A and B in figure 9-4-1 are explained as follows.

■ Setup Procedures at A

1. Set the ATC memory mode register (ATnMMD, ATnMMDS).

2. Set the ATC interrupt register (ATIRQ).
Set the interrupt enable flag of the corresponded channel.

3. Set the ATC transfer word count register (ATnCNT).
The 16 byte data is 8 words in word transfer. Therefore, write x'0007' (x'08'-1) to the ATnCNT register.

4. Set the ATC source address register (ATnSRC).
Write x'200000' to the ATnSRC register.

5. Set the ATC destination address register (ATnDST).
Write x'600000' to the ATnDST register.

6. Set the ATC control register (ATnCTR).
Write x'3205' to the ATnCTR register.

Activation factor:	hardware activation (ATnIQ=1)
Source increment mode:	increment (ATnSI[1:0]=01)
Destination increment mode:	fixed (ATDI[1:0]=00)
Addressing mode:	dual (ATnAMD[1:0]=00)
Transfer unit:	word (ATnTBW=1)
Transfer mode:	burst transfer (ATnTMD=1)
Transfer enable:	enable (ATnEN=1)
ATnEN invalid:	ATnEN valid (ATnENX=0)
Transfer start and busy flag:	either way (disable to write)

■ Setup Procedures at B

1. Set the ATC interrupt register (ATIRQ).

Set the interrupt enable flag (ATnIR) of the corresponded channel. At this point, an interrupt request to the CPU becomes disabled.

2. Set the ATC transfer word count register (ATnCNT).

Set the value only when the set value needs to be changed. (Reload the preset value when the transfer is completed.)

3. Set the ATC source address register (ATnSRC).

at the value only when the set value needs to be changed. (Write all 3 bytes.)

4. Set the ATC control register (ATnCTR).

Because the transfer enable flag is 0, rewrite x'3205' to the ATnCTR register.

Activation factor:	hardware activation (ATnIQ=1)
Source increment mode:	increment (ATnSI[1:0]=01)
Destination increment mode:	fixed (ATDI[1:0]=00)
Addressing mode:	dual (ATnAMD[1:0]=00)
Transfer unit:	word (ATnTBW=1)
Transfer mode:	burst transfer (ATnTMD=1)
Transfer enable:	enable (ATnEN=1)
ATnEN invalid:	ATnEN valid (ATnENX=0)
Transfer start and busy flag:	either way (disable to write)

Chapter 10 System Control

10-1 Summary of Address Break Function

10-1-1 Address Break

The MN102H74G/74F/74D/F74G generates a NMI interrupt before executing the instruction allocated on any address. The MN102H74G/74F/74D/F74G can specify addresses where a NMI interrupt occurs in up to two registers of address break 0 address pointer and address break 1 address pointer. When the instruction fetch address matches to one of address break address pointers, a NMI interrupt occurs by replacing the instruction code of the address to the undefined instruction (x'FF'). The address break function is used as the software debugger as well as ROM correction in Mask ROM version.

> This function is not able to used in ICE.

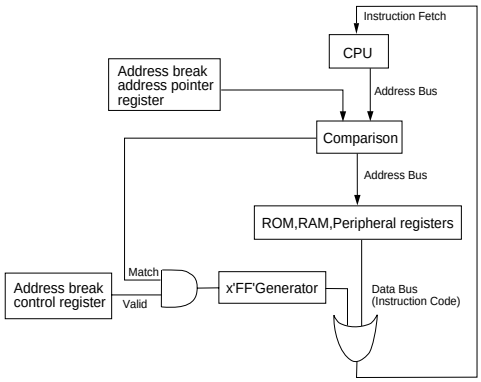


Figure 10-1-1 Address Break Block Diagram

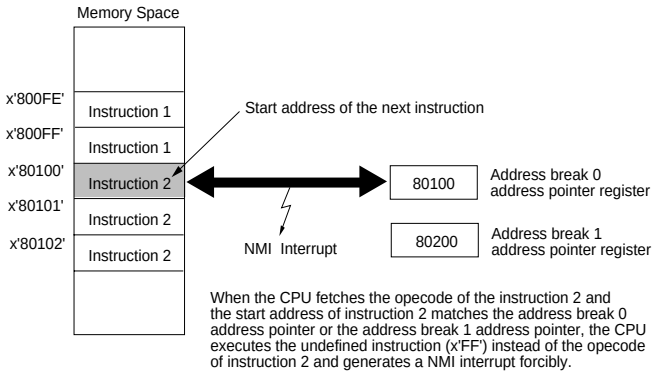

Memory Space
x'800FE'
Instruction 1
x'800FF'
Instruction 1
x'80100'
Instruction 2
x'80101'
Instruction 2
x'80102'
Instruction 2
Start address of the next instruction
NMI Interrupt
80100
Address break 0 address pointer register
80200
Address break 1 address pointer register
When the CPU fetches the opecode of the instruction 2 and the start address of instruction 2 matches the address break 0 address pointer or the address break 1 address pointer, the CPU executes the undefined instruction (x'FF') instead of the opecode of instruction 2 and generates a NMI interrupt forcibly.

Figure 10-1-2 Address Break Operation Example

> Set the start address of the instruction code to be suspended in the address break address pointer.

> All instructions can be suspended.

10-1-2 Control Registers

The address break n address pointers and the address break control register control address break function.

	7	6	5	4	3	2	1	0
ADBCTL	—	—	—	—	ADB1ON	ADB0ON	ADB1ST	ADB0ST
Reset:	0	0	0	0	0	0	0	0

ADB0ST	Address Break 0 Generation
0	Not generated
1	Generated

ADB1ST	Address Break 1 Generation
0	Not generated
1	Generated

ADB0ON	Address Break 0 On/Off
0	OFF
1	ON

ADB1ON	Address Break 1 On/Off
0	OFF
1	ON

Figure 10-1-3 Address Break Control Register
(ADBCTL:x'E0050A' Byte Access/Word Access, R/W)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADBn	ADBnA15	ADBnA14	ADBnA13	ADBnA12	ADBnA11	ADBnA10	ADBnA9	ADBnA8	ADBnA7	ADBnA6	ADBnA5	ADBnA4	ADBnA3	ADBnA2	ADBnA1	ADBnA0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADBnH	-	-	-	-	-	-	-	-	ADBnA23	ADBnA22	ADBnA21	ADBnA20	ADBnA19	ADBnA18	ADBnA17	ADBnA16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-1-4 Address Break n Address Pointer
(ADB0:x'E00502' ADB1:x'E00506' Byte Access/Word Access, R/W)

10-1-3 Address Break Setup Example

When an error occurs on the routine on internal ROM, the error can be avoided by storing its solution program on Internal RAM and setting the address break function. For example, the address for the address break and the replace program are stored on nonvolatile memory externally connected to the chip

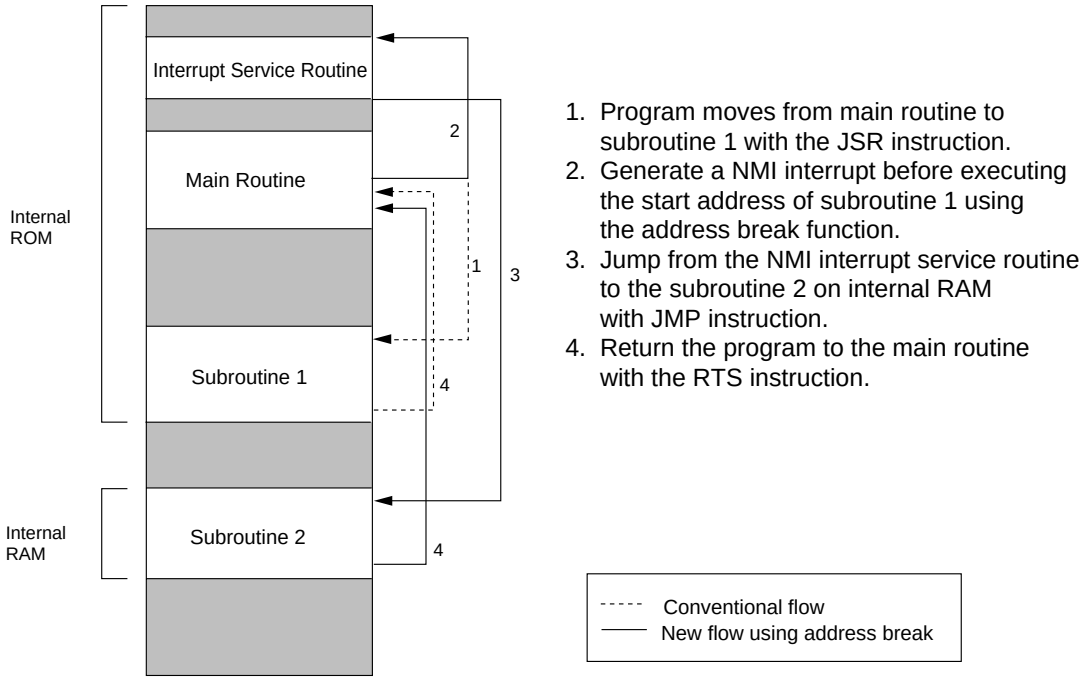

Interrupt Service Routine
Main Routine
Subroutine 1
Subroutine 2
Internal ROM
Internal RAM
1
2
3
4
4
1. Program moves from main routine to subroutine 1 with the JSR instruction.
2. Generate a NMI interrupt before executing the start address of subroutine 1 using the address break function.
3. Jump from the NMI interrupt service routine to the subroutine 2 on internal RAM with JMP instruction.
4. Return the program to the main routine with the RTS instruction.
Conventional flow
New flow using address break

Figure 10-1-5 Address Break Setup Flow

■ Address Break Setup

(1) Set the start address of subroutine 1 to the address break 0 address pointer (ADB0).

(2) Set the ADB0ON flag of the address break control register to 1.

ADBCTL: x'E0050A'

7	6	5	4	3	2	1	0
-	-	-	-	ADB1ON	ADB0ON	ADB1ST	ADB0ST
0	0	0	0	0	1	0	0

Thereafter, a NMI interrupt occurs when the CPU executes the address set in step (1).

■ NMI Interrupt Service Routine

(3) When the address break occurs, the program jumps to the address x'80008'. The IAGR register value is 8 at this point. Verify that the ADB0ST flag of the ADBCTL register is 1 while the NMI interrupt service routine is executed. This verification identifies whether a NMI interrupt occurs using address break or another factor. Clear the ADB0ST flag and the ADB1ST flag to 0 using software because those flags cannot be cleared automatically.

ADBCTL: x'E0050A'

7	6	5	4	3	2	1	0
-	-	-	-	ADB1ON	ADB0ON	ADB1ST	ADB0ST
0	0	0	0	0	1	0	1

(4) The program jumps to the subroutine 2 on internal RAM in advance. At this point, add the stack pointer (A3 register) value to 6 to delete the program counter (PC) value stored by a NMI interrupt and the PSW value from the stack. On addition, clear the NMIF flag of the G0ICR register to 0.

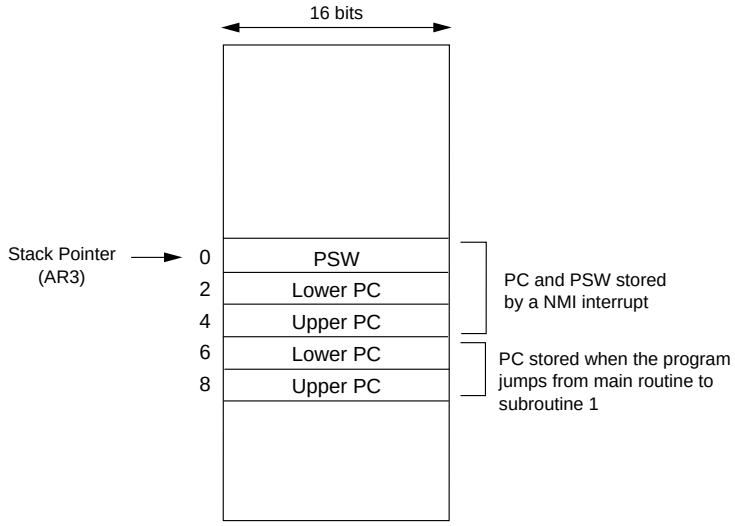


Figure 10-1-6 Stack State after NMI Interrupt

(5) Execute the subroutine 2 on internal RAM and return the main routine with the RTS instruction.

10-2 System-related Register Protection

10-2-1 Overview

The MN102H74G/74F/74D/F74G contains the system control register to protect the overwrite of the system-related registers due to errors. Setting the values except x'7D' to the system control register prohibits the overwrite of the system-related registers.

10-2-2 System Control Register

	7	6	5	4	3	2	1	0
SYSCTL	SYSC7	SYSC6	SYSC5	SYSC4	SYSC3	SYSC2	SYSC1	SYSC0
Reset:	0	1	1	1	1	1	0	1

7D: Enable the write of all registers
Others: Disable the write of the following registers

CPU Control Registers:	CPUM,EFCR
Address Break Control Registers:	ADB0,ADB1,ADBCTL
Port Control Registers:	P01MD,P1MD,P2MD, P3MD,P4LMD,P4HMD, P5MD,P6MD,P7MD, P8MD,P9MD,PALMD,PAHMD, P0DIR,P1DIR,P2DIR, P3DIR,P4DIR,P5DIR, P6DIR,P7DIR,P8DIR P9DIR,PADIR,PBDIR

Figure 10-2-1 System Control Register (SYSCTL: x'E00500')

Chapter 11 Ports

11

11-1 Summary of Ports

11-1-1 Overview

The MN102H74G/74F/74D/F74G contains 12 groups of I/O ports. Refer to the pin table because functions can be switched depending on the selected mode pins.

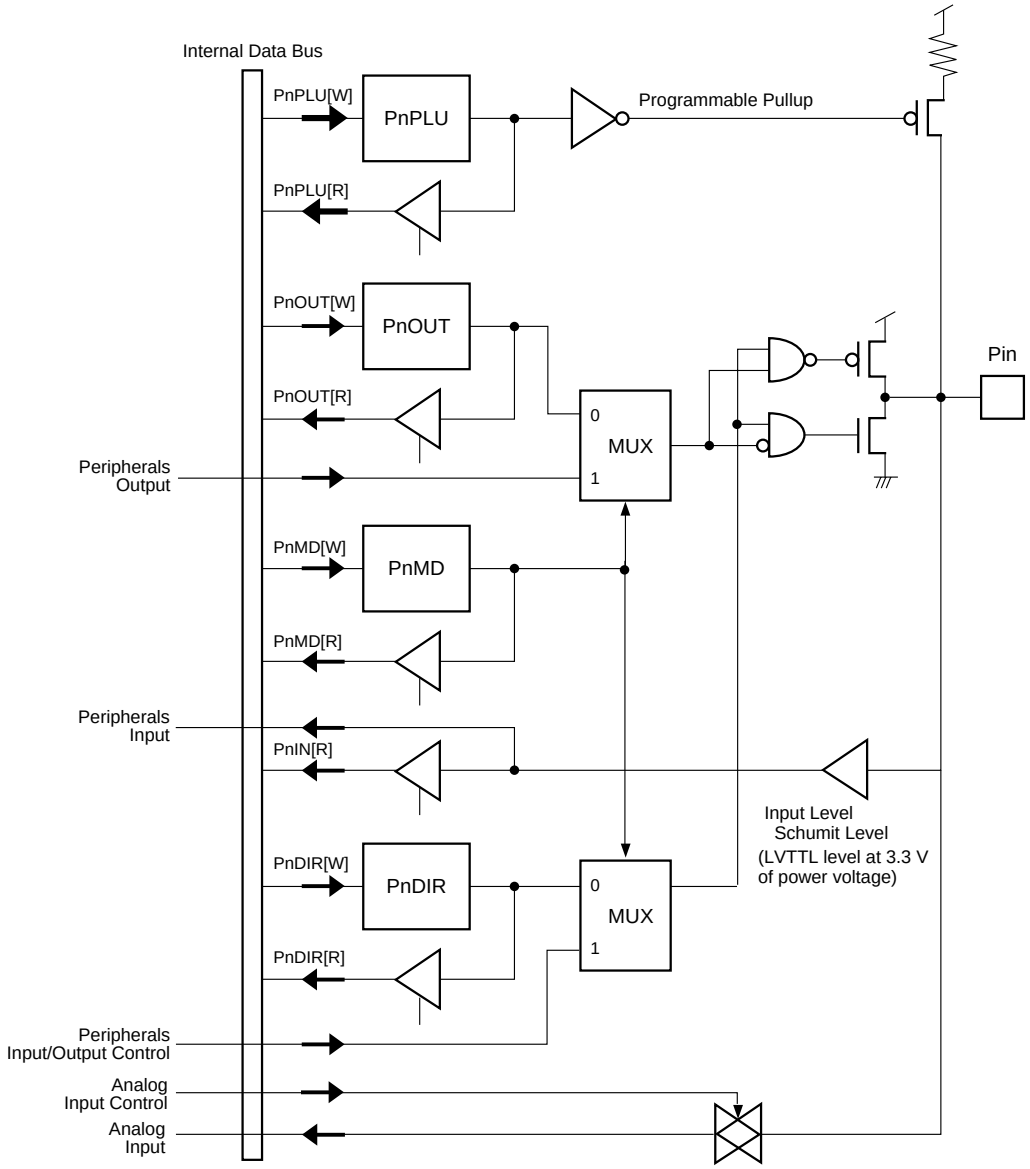


Figure 11-1-1 I/O Port Configuration

11-2 Control Registers

These registers control ports: the port output register (PnOUT), the port input register (PnIN), the port mode register (PnMD), the port input/output control register (PnDIR) and the port pull-up control register (PnPLU). Refer to “12-3 List of Pins” for the details of implemented bits. The port input/output control register determines the port direction only when each port is used as a port input/output function. The port mode register determines the port direction when each port is used as an input/output pin of peripheral function.

7	6	5	4	3	2	1	0	
PnOUT7	PnOUT6	PnOUT5	PnOUT4	PnOUT3	PnOUT2	PnOUT1	PnOUT0	PnOUT

7	6	5	4	3	2	1	0	
PnIN7	PnIN6	PnIN5	PnIN4	PnIN3	PnIN2	PnIN1	PnIN0	PnIN

7	6	5	4	3	2	1	0		
PnMD7	PnMD6	PnMD5	PnMD4	PnMD3	PnMD2	PnMD1	PnMD0	PnMD	PnLMD PnHMD

7	6	5	4	3	2	1	0		
PnDIR7	PnDIR6	PnDIR5	PnDIR4	PnDIR3	PnDIR2	PnDIR1	PnDIR0	PnDIR	0 : Input 1 : Output

7	6	5	4	3	2	1	0		
PnPLU7	PnPLU6	PnPLU5	PnPLU4	PnPLU3	PnPLU2	PnPLU1	PnPLU0	PnPLU	0 : Pull-up Off 1 : Pull-up On

11-2-1 List of Port Control Registers

Table 11-2-1 List of Port Control Registers (1/2)

Register	Adress	R/W	Function
P0PLU	x'E00700'	R/W	Port 0 Pull-up Control Register
P0OUT	x'E00710'	R/W	Port 0 Output Register
P0IN	x'E00720'	R	Port 0 Input Register
P0DIR	x'E00730'	R/W	Port 0 Input/Output Control Register
P01MD	x'E00740'	R/W	Port 0, 1 Mode Register
P1PLU	x'E00701'	R/W	Port 1 Pull-up Control Register
P1OUT	x'E00711'	R/W	Port 1 Output Register
P1IN	x'E00721'	R	Port 1 Input Register
P1DIR	x'E00731'	R/W	Port 1 Input/Output Control Register
P1MD	x'E00741'	R/W	Port 1 Mode Register
P2PLU	x'E00702'	R/W	Port 2 Pull-up Control Register
P2OUT	x'E00712'	R/W	Port 2 Output Register
P2IN	x'E00722'	R	Port 2 Input Register
P2DIR	x'E00732'	R/W	Port 2 Input/Output Control Register
P2MD	x'E00742'	R/W	Port 2 Mode Register
P3PLU	x'E00703'	R/W	Port 3 Pull-up Control Register
P3OUT	x'E00713'	R/W	Port 3 Output Register
P3IN	x'E00723'	R	Port 3 Input Register
P3DIR	x'E00733'	R/W	Port 3 Input/Output Control Register
P3MD	x'E00743'	R/W	Port 3 Mode Register
P4PLU	x'E00704'	R/W	Port 4 Pull-up Control Register
P4OUT	x'E00714'	R/W	Port 4 Output Register
P4IN	x'E00724'	R	Port 4 Input Register
P4DIR	x'E00734'	R/W	Port 4 Input/Output Control Register
P4LMD	x'E00744'	R/W	Port 4 Mode Register L
P4HMD	x'E00745'	R/W	Port 4 Mode Register H
P5PLU	x'E00705'	R/W	Port 5 Pull-up Control Register
P5OUT	x'E00715'	R/W	Port 5 Output Register
P5IN	x'E00725'	R	Port 5 Input Register
P5DIR	x'E00735'	R/W	Port 5 Input/Output Control Register
P5MD	x'E00746'	R/W	Port 5 Mode Register
P6PLU	x'E00706'	R/W	Port 6 Pull-up Control Register
P6OUT	x'E00716'	R/W	Port 6 Output Register
P6IN	x'E00726'	R	Port 6 Input Register
P6DIR	x'E00736'	R/W	Port 6 Input/Output Control Register
P6MD	x'E00747'	R/W	Port 6 Mode Register

Table 11-2-1 List of Port Control Registers (2/2)

P7PLU	x'E00707'	R/W	Port 7 Pull-up Control Register
P7OUT	x'E00717'	R/W	Port 7 Output Register
P7IN	x'E00727'	R	Port 7 Input Register
P7DIR	x'E00737'	R/W	Port 7 Input/Output Control Register
P7MD	x'E00748'	R/W	Port 7 Mode Register
P8PLU	x'E00708'	R/W	Port 8 Pull-up Control Register
P8OUT	x'E00718'	R/W	Port 8 Output Register
P8IN	x'E00728'	R	Port 8 Input Register
P8DIR	x'E00738'	R/W	Port 8 Input/Output Control Register
P8MD	x'E00749'	R/W	Port 8 Mode Register
P9PLU	x'E00709'	R/W	Port 9 Pull-up Control Register
P9OUT	x'E00719'	R/W	Port 9 Output Register
P9IN	x'E00729'	R	Port 9 Input Register
P9DIR	x'E00739'	R/W	Port 9 Input/Output Control Register
P9MD	x'E0074A'	R/W	Port 9 Mode Register
PAPLU	x'E0070A'	R/W	Port A Pull-up Control Register
PAOUT	x'E0071A'	R/W	Port A Output Register
PAIN	x'E0072A'	R	Port A Input Register
PADIR	x'E0073A'	R/W	Port A Input/Output Control Register
PALMD	x'E0074C'	R/W	Port A Mode Register L
PAHMD	x'E0074D'	R/W	Port A Mode Register H
PBPLU	x'E0070B'	R/W	Port B Pull-up Control Register
PBOUT	x'E0071B'	R/W	Port B Output Register
PBIN	x'E0072B'	R	Port B Input Register
PBDIR	x'E0073B'	R/W	Port B Input/Output Control Register

11-2-2 Register of Port 0

	7	6	5	4	3	2	1	0
P0PLU	P0 PLU7	P0 PLU6	P0 PLU5	P0 PLU4	P0 PLU3	P0 PLU2	P0 PLU1	P0 PLU0
Reset:	0	0	0	0	0	0	0	0

P0PLUn(n=7-0)	Port 0 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 0 pull-up register (P0PLU : x'E00700' R/W)

	7	6	5	4	3	2	1	0
P0OUT	P0 OUT7	P0 OUT6	P0 OUT5	P0 OUT4	P0 OUT3	P0 OUT2	P0 OUT1	P0 OUT0
Reset:	0	0	0	0	0	0	0	0

P0OUTn(n=7-0)	Port 0 Output Select
0	Low
1	High

Port 0 output register (P0OUT : x'E00710' R/W)

	7	6	5	4	3	2	1	0
P0IN	P0 IN7	P0 IN6	P0 IN5	P0 IN4	P0 IN3	P0 IN2	P0 IN1	P0 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

P0INn(n=7-0)	Port 0 Input
0	Low
1	High

Port 0 input register (P0IN : x'E00720' R)

	7	6	5	4	3	2	1	0
P0DIR	P0 DIR7	P0 DIR6	P0 DIR5	P0 DIR4	P0 DIR3	P0 DIR2	P0 DIR1	P0 DIR0
Reset:	0	0	0	0	0	0	0	0

P0DIRn(n=7-0)	Port 0 Input/Output Direction Select
0	Input
1	Output

Port 0 input/output control register (P0DIR : x'E00730' R/W)

	7	6	5	4	3	2	1	0	
P01MD	—	—	—	—	—	P01 MD2	P01 MD1	P01 MD0	
Reset:	0	0	0	0	0	0	1	1	:Processor Mode (16-bit Bus Width)
	0	0	0	0	0	0	0	1	:Processor Mode (8-bit Bus Width)
	0	0	0	0	0	0	0	0	:Single-chip Mode/Memory Expansion Mode

P01MD0	Port 0 Input/Output Signal Select
0	Port
1	Data

P01MD2	P01MD1	Port 1 Input/Output Signal Select (Note)
0	0	Port
0	1	Data
1	0	Timer

Note: Do not use timer input function of port 4 and port 7 when using port 1 as timer function.

Reserved: Set 0.

Port 0, 1 mode register (P01MD : x'E00740' R/W)

11-2-3 Register of Port 1

	7	6	5	4	3	2	1	0
P1PLU	P1 PLU7	P1 PLU6	P1 PLU5	P1 PLU4	P1 PLU3	P1 PLU2	P1 PLU1	P1 PLU0
Reset:	0	0	0	0	0	0	0	0

P1PLUn(n=7-0)	Port 1 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 1 pull-up register (P1PLU : x'E00701' R/W)

	7	6	5	4	3	2	1	0
P1OUT	P1 OUT7	P1 OUT6	P1 OUT5	P1 OUT4	P1 OUT3	P1 OUT2	P1 OUT1	P1 OUT0
Reset:	0	0	0	0	0	0	0	0

P1OUTn(n=7-0)	Port 1 Output Select
0	Low
1	High

Port 1 output register (P1OUT : x'E00711' R/W)

	7	6	5	4	3	2	1	0
P1IN	P1 IN7	P1 IN6	P1 IN5	P1 IN4	P1 IN3	P1 IN2	P1 IN1	P1 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

P1INn(n=7-0)	Port 1 Input
0	Low
1	High

Port 1 input register (P1IN : x'E00721' R)

	7	6	5	4	3	2	1	0
P1DIR	P1 DIR7	P1 DIR6	P1 DIR5	P1 DIR4	P1 DIR3	P1 DIR2	P1 DIR1	P1 DIR0
Reset:	0	0	0	0	0	0	0	0

P1DIRn(n=7-0)	Port 1 Input/Output Direction Select
0	Input
1	Output

Port 1 input/output control register (P1DIR : x'E00731' R/W)

	7	6	5	4	3	2	1	0
P1MD	P1 MD7	P1 MD6	P1 MD5	P1 MD4	P1 MD3	P1 MD2	P1 MD1	P1 MD0
Reset:	0	0	0	0	0	0	0	0

P1MD0	P10 Input/Output Signal Select
0	Port/TM2IO input
1	TM2IO output

P1MD1	P11 Input/Output Signal Select
0	Port/TM3IO input
1	TM3IO output

P1MD2	P12 Input/Output Signal Select
0	Port/TM4IO input
1	TM4IO output

P1MD3	P13 Input/Output Signal Select
0	Port/TM5IO input
1	TM5IO output

P1MD4	P14 Input/Output Signal Select
0	Port/TM6IO input
1	TM6IO output

P1MD5	P15 Input/Output Signal Select
0	Port/TM7IO input
1	TM7IO output

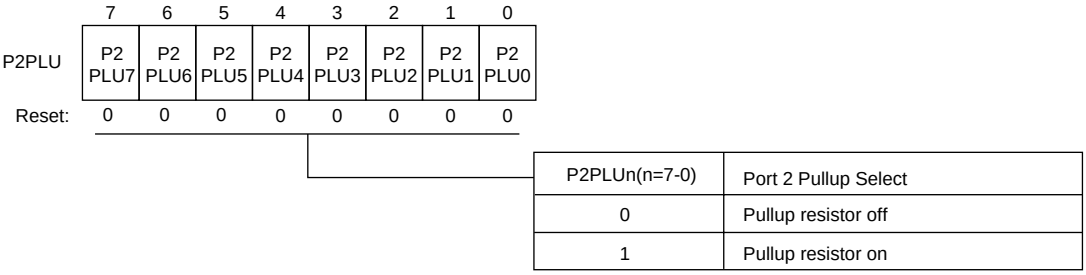
P1MD6	P16 Input/Output Signal Select
0	Port/TM8IO input
1	TM8IO output

P1MD7	P17 Input/Output Signal Select
0	Port/TM9IO input
1	TM9IO output

The P01MD1 must be set to 0 when the port 1 is used as a port or a peripheral function.

Port 1 mode register (P1MD : x'E00741' R/W)

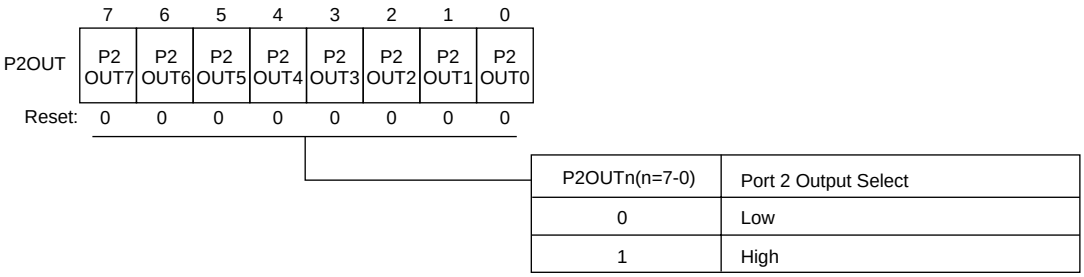
11-2-4 Register of Port 2



P2PLU	7	6	5	4	3	2	1	0
	P2 PLU7	P2 PLU6	P2 PLU5	P2 PLU4	P2 PLU3	P2 PLU2	P2 PLU1	P2 PLU0
Reset:	0	0	0	0	0	0	0	0

P2PLUn(n=7-0)	Port 2 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

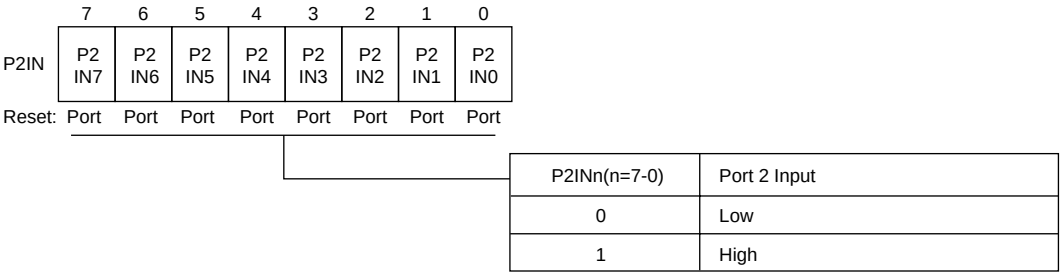
Port 2 pull-up register (P2PLU : x'E00702' R/W)



P2OUT	7	6	5	4	3	2	1	0
	P2 OUT7	P2 OUT6	P2 OUT5	P2 OUT4	P2 OUT3	P2 OUT2	P2 OUT1	P2 OUT0
Reset:	0	0	0	0	0	0	0	0

P2OUTn(n=7-0)	Port 2 Output Select
0	Low
1	High

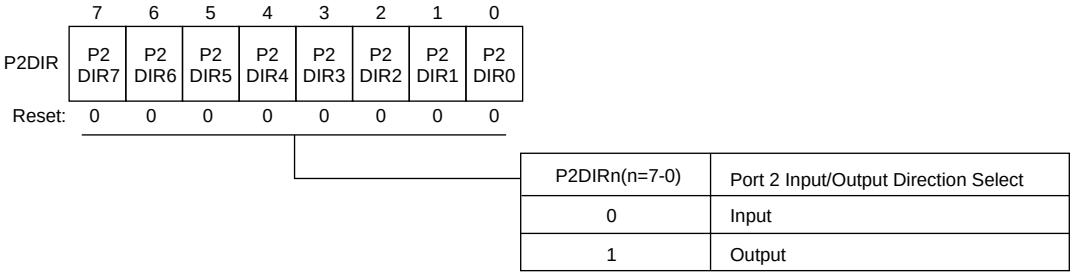
Port 2 output register (P2OUT : x'E00712' R/W)



P2IN	7	6	5	4	3	2	1	0
	P2 IN7	P2 IN6	P2 IN5	P2 IN4	P2 IN3	P2 IN2	P2 IN1	P2 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

P2INn(n=7-0)	Port 2 Input
0	Low
1	High

Port 2 input register (P2IN : x'E00722' R)



P2DIR	7	6	5	4	3	2	1	0
	P2 DIR7	P2 DIR6	P2 DIR5	P2 DIR4	P2 DIR3	P2 DIR2	P2 DIR1	P2 DIR0
Reset:	0	0	0	0	0	0	0	0

P2DIRn(n=7-0)	Port 2 Input/Output Direction Select
0	Input
1	Output

Port 2 input/output control register (P2DIR : x'E00732' R/W)

	7	6	5	4	3	2	1	0
P2MD	P2 MD7	P2 MD6	P2 MD5	P2 MD4	P2 MD3	P2 MD2	P2 MD1	P2 MD0
Reset:	1	1	1	1	1	1	1	1 :Processor Mode
	0	0	0	0	0	0	0	0 :Single-chip Mode/Memory Expansion Mode

P2MD0	P20 Input/Output Signal Select
0	Port
1	A00

P2MD1	P21 Input/Output Signal Select
0	Port
1	A01

P2MD2	P22 Input/Output Signal Select
0	Port
1	A02

P2MD3	P23 Input/Output Signal Select
0	Port
1	A03

P2MD4	P24 Input/Output Signal Select
0	Port
1	A04

P2MD5	P25 Input/Output Signal Select
0	Port
1	A05

P2MD6	P26 Input/Output Signal Select
0	Port
1	A06

P2MD7	P27 Input/Output Signal Select
0	Port
1	A07

Port 2 mode register (P2MD : x'E00742' R/W)

11-2-5 Register of Port 3

	7	6	5	4	3	2	1	0
P3PLU	P3 PLU7	P3 PLU6	P3 PLU5	P3 PLU4	P3 PLU3	P3 PLU2	P3 PLU1	P3 PLU0
Reset:	0	0	0	0	0	0	0	0

P3PLUn(n=7-0)	Port 3 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 3 pull-up register (P3PLU : x'E00703' R/W)

	7	6	5	4	3	2	1	0
P3OUT	P3 OUT7	P3 OUT6	P3 OUT5	P3 OUT4	P3 OUT3	P3 OUT2	P3 OUT1	P3 OUT0
Reset:	0	0	0	0	0	0	0	0

P3OUTn(n=7-0)	Port 3 Output Select
0	Low
1	High

Port 3 output register (P3OUT : x'E00713' R/W)

	7	6	5	4	3	2	1	0
P3IN	P3 IN7	P3 IN6	P3 IN5	P3 IN4	P3 IN3	P3 IN2	P3 IN1	P3 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

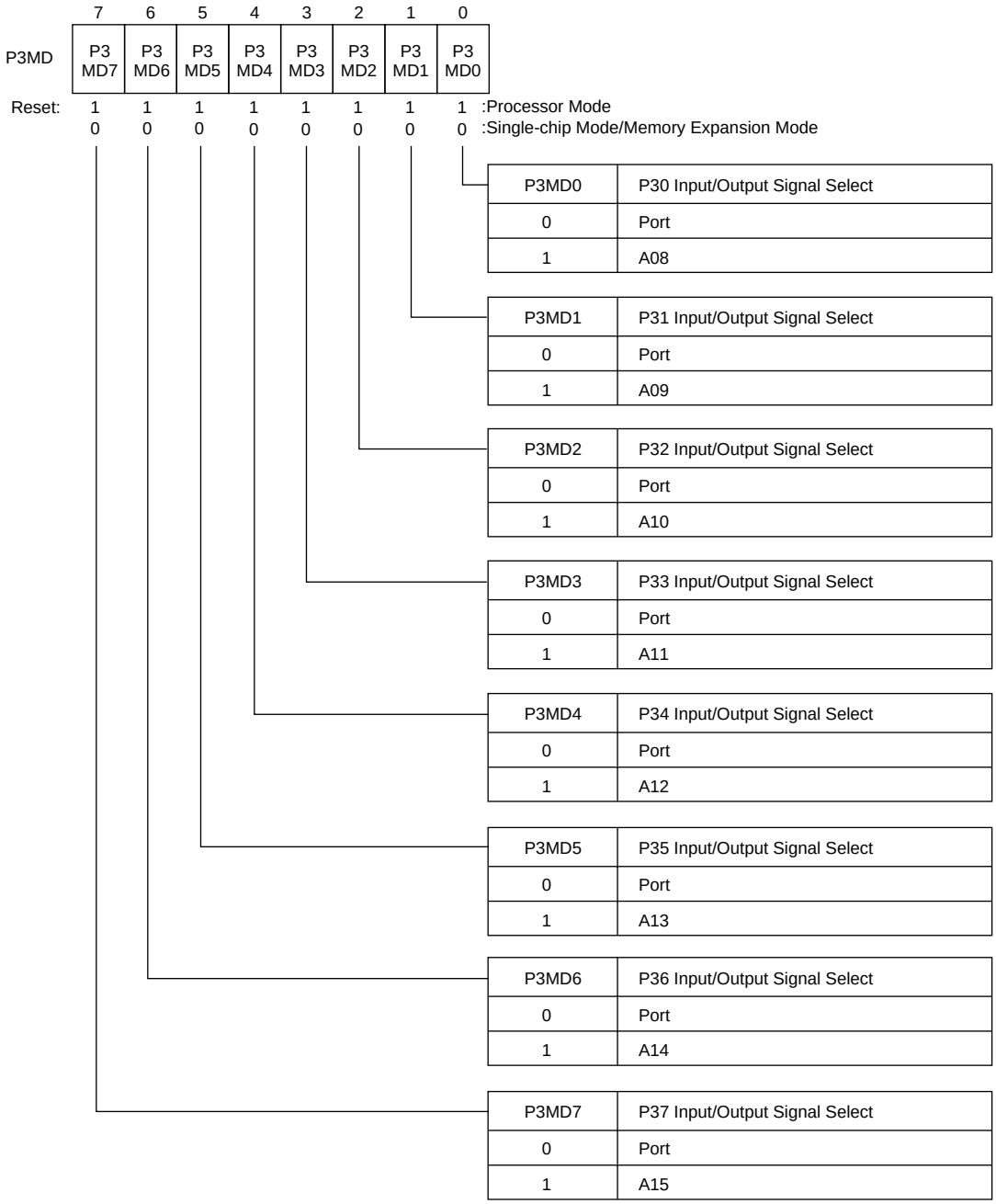
P3INn(n=7-0)	Port 3 Input
0	Low
1	High

Port 3 input register (P3IN : x'E00723' R)

	7	6	5	4	3	2	1	0
P3DIR	P3 DIR7	P3 DIR6	P3 DIR5	P3 DIR4	P3 DIR3	P3 DIR2	P3 DIR1	P3 DIR0
Reset:	0	0	0	0	0	0	0	0

P3DIRn(n=7-0)	Port 3 Input/Output Direction Select
0	Input
1	Output

Port 3 input/output control register (P3DIR : x'E00733' R/W)



	7	6	5	4	3	2	1	0	
P3MD	P3 MD7	P3 MD6	P3 MD5	P3 MD4	P3 MD3	P3 MD2	P3 MD1	P3 MD0	
Reset:	1	1	1	1	1	1	1	1	:Processor Mode
	0	0	0	0	0	0	0	0	:Single-chip Mode/Memory Expansion Mode

P3MD0	P30 Input/Output Signal Select
0	Port
1	A08

P3MD1	P31 Input/Output Signal Select
0	Port
1	A09

P3MD2	P32 Input/Output Signal Select
0	Port
1	A10

P3MD3	P33 Input/Output Signal Select
0	Port
1	A11

P3MD4	P34 Input/Output Signal Select
0	Port
1	A12

P3MD5	P35 Input/Output Signal Select
0	Port
1	A13

P3MD6	P36 Input/Output Signal Select
0	Port
1	A14

P3MD7	P37 Input/Output Signal Select
0	Port
1	A15

Port 3 mode register (P3MD : x'E00743' R/W)

11-2-6 Register of Port 4

	7	6	5	4	3	2	1	0
P4PLU	P4 PLU7	P4 PLU6	P4 PLU5	P4 PLU4	P4 PLU3	P4 PLU2	P4 PLU1	P4 PLU0
Reset:	0	0	0	0	0	0	0	0

P4PLUn(n=7-0)	Port 4 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 4 pull-up register (P4PLU : x'E00704' R/W)

	7	6	5	4	3	2	1	0
P4OUT	P4 OUT7	P4 OUT6	P4 OUT5	P4 OUT4	P4 OUT3	P4 OUT2	P4 OUT1	P4 OUT0
Reset:	0	0	0	0	0	0	0	0

P4OUTn(n=7-0)	Port 4 Output Select
0	Low
1	High

Port 4 output register (P4OUT : x'E00714' R/W)

	7	6	5	4	3	2	1	0
P4IN	P4 IN7	P4 IN6	P4 IN5	P4 IN4	P4 IN3	P4 IN2	P4 IN1	P4 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

P4INn(n=7-0)	Port 4 Input
0	Low
1	High

Port 4 input register (P4IN : x'E00724' R)

	7	6	5	4	3	2	1	0
P4DIR	P4 DIR7	P4 DIR6	P4 DIR5	P4 DIR4	P4 DIR3	P4 DIR2	P4 DIR1	P4 DIR0
Reset:	0	0	0	0	0	0	0	0

P4DIRn(n=7-0)	Port 4 Input/Output Direction Select
0	Input
1	Output

Port 4 input/output control register (P4DIR : x'E00734' R/W)

	7	6	5	4	3	2	1	0	
P4LMD	—	—	—	P4 LMD4	P4 LMD3	P4 LMD2	P4 LMD1	P4 LMD0	
Reset:	0	0	0	0	1	1	1	1	:Processor Mode
	0	0	0	0	0	0	0	0	:Single-chip Mode/Memory Expansion Mode

P4LMD0	P40 Input/Output Signal Select
0	Port
1	A16

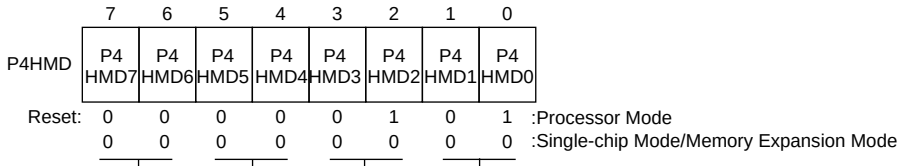
P4LMD1	P41 Input/Output Signal Select
0	Port
1	A17

P4LMD2	P42 Input/Output Signal Select
0	Port
1	A18

P4LMD4	P4LMD3	P43 Input/Output Signal Select
0	0	Port/TM2IO input
0	1	A19
1	0	TM2IO output

Note: Do not select timer input when using Port 1 as timer.

Port 4 mode register L (P4LMD : x'E00744' R/W)


7 6 5 4 3 2 1 0
P4HMD
P4 HMD7 P4 HMD6 P4 HMD5 P4 HMD4 P4 HMD3 P4 HMD2 P4 HMD1 P4 HMD0
Reset: 0 0 0 0 0 1 0 1 :Processor Mode
0 0 0 0 0 0 0 0 :Single-chip Mode/Memory Expansion Mode

P4HMD1	P4HMD0	P44 Input/Output Signal Select
0	0	Port/TM3IO input
0	1	A20
1	0	TM3IO output

P4HMD3	P4HMD2	P45 Input/Output Signal Select
0	0	Port/TM4IO input
0	1	A21
1	0	TM4IO output

P4HMD5	P4HMD4	P46 Input/Output Signal Select
0	0	Port/TM5IO input
0	1	A22
1	0	TM5IO output

P4HMD7	P4HMD6	P47 Input/Output Signal Select
0	0	Port/TM6IO input
0	1	A23
1	0	TM6IO output
1	1	/CS0S

Note: Do not select timer input when using Port 1 as timer.

Port 4 mode register H (P4HMD : x'E00745' R/W)

11-2-7 Register of Port 5

	7	6	5	4	3	2	1	0
P5PLU	—	—	—	—	P5 PLU3	P5 PLU2	P5 PLU1	P5 PLU0
Reset:	0	0	0	0	0	0	0	0

P5PLUn(n=3-0)	Port 5 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 5 pull-up register (P5PLU : x'E00705' R/W)

	7	6	5	4	3	2	1	0
P5OUT	—	—	—	P5 OUT4	P5 OUT3	P5 OUT2	P5 OUT1	P5 OUT0
Reset:	0	0	0	0	0	0	0	0

P5OUTn(n=4-0)	Port 5 Output Select
0	Low
1	High

Port 5 output register (P5OUT : x'E00715' R/W)

	7	6	5	4	3	2	1	0
P5IN	—	—	—	—	P5 IN3	P5 IN2	P5 IN1	P5 IN0
Reset:	0	0	0	0	Port	Port	Port	Port

P5INn(n=3-0)	Port 5 Input
0	Low
1	High

Port 5 input register (P5IN : x'E00725' R)

	7	6	5	4	3	2	1	0
P5DIR	—	—	—	—	P5 DIR3	P5 DIR2	P5 DIR1	P5 DIR0
Reset:	0	0	0	0	0	0	0	0

P5DIRn(n=3-0)	Port 5 Input/Output Direction Select
0	Input
1	Output

Port 5 input/output control register (P5DIR : x'E00735' R/W)

	7	6	5	4	3	2	1	0	
P5MD	—	—	—	P5 MD4	P5 MD3	P5 MD2	P5 MD1	P5 MD0	
Reset:	0	0	0	0	0	1	1	1	:Processor Mode (16-bit Bus Width)
	0	0	0	0	0	0	1	1	:Processor Mode (8-bit Bus Width)
	0	0	0	0	0	0	0	0	:Single-chip Mode/Memory Expansion Mode

P5MD0	P51 Input/Output Signal Select
0	Port
1	/RE

P5MD1	P52 Input/Output Signal Select
0	Port
1	/WEL

P5MD2	P53 Input/Output Signal Select
0	Port
1	/WEH

P5MD4	P5MD3	P54 Input/Output Signal Select
0	0	BOSC
0	1	SYSCLK
1	0	Port output

Port 5 mode register (P5MD : x'E00746' R/W)

P50 pin is used as WAIT signal pin as well, but because the WAIT signal pin is only for input, so when reset is released, P50 pin can be used in port input mode. In handshake mode, it can used for the WAIT signal by setting of MEMMDn and MEMMDnS register.

11-2-8 Register of Port 6

	7	6	5	4	3	2	1	0
P6PLU	P6 PLU7	P6 PLU6	P6 PLU5	P6 PLU4	P6 PLU3	P6 PLU2	P6 PLU1	P6 PLU0
Reset:	0	0	0	0	0	0	0	0

P6PLUn(n=7-0)	Port 6 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 6 pull-up register (P6PLU : x'E00706' R/W)

	7	6	5	4	3	2	1	0
P6OUT	P6 OUT7	P6 OUT6	P6 OUT5	P6 OUT4	P6 OUT3	P6 OUT2	P6 OUT1	P6 OUT0
Reset:	0	0	0	0	0	0	0	0

P6OUTn(n=7-0)	Port 6 Output Select
0	Low
1	High

Port 6 output register (P6OUT : x'E00716' R/W)

	7	6	5	4	3	2	1	0
P6IN	P6 IN7	P6 IN6	P6 IN5	P6 IN4	P6 IN3	P6 IN2	P6 IN1	P6 IN0
Reset:	Port	Port	Port	Port	Port	Port	Port	Port

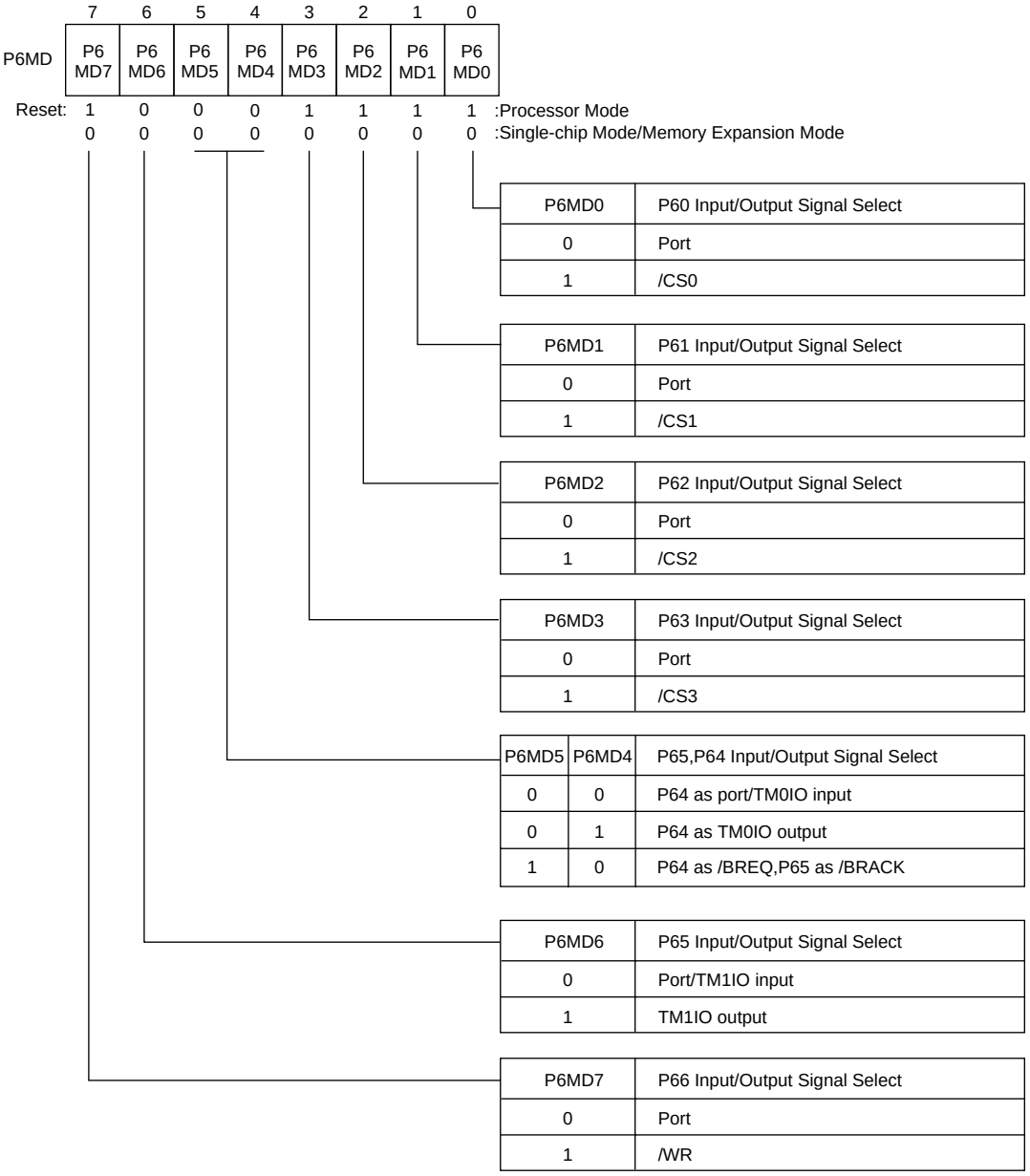
P6INn(n=7-0)	Port 6 Input
0	Low
1	High

Port 6 input register (P6IN : x'E00726' R)

	7	6	5	4	3	2	1	0
P6DIR	P6 DIR7	P6 DIR6	P6 DIR5	P6 DIR4	P6 DIR3	P6 DIR2	P6 DIR1	P6 DIR0
Reset:	0	0	0	0	0	0	0	0

P6DIRn(n=7-0)	Port 6 Input/Output Direction Select
0	Input
1	Output

Port 6 input/output control register (P6DIR : x'E00736' R/W)



Port 6 mode register (P6MD : x'E00747' R/W)

P67 pin is used as /WORD signal as well in only single-chip mode.

11-2-9 Register of Port 7

	7	6	5	4	3	2	1	0
P7PLU	—	—	—	—	—	P7 PLU2	P7 PLU1	P7 PLU0
Reset:	0	0	0	0	0	0	0	0

P7PLUn(n=2-0)	Port 7 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 7 pull-up register (P7PLU : x'E00707' R/W)

	7	6	5	4	3	2	1	0
P7OUT	—	—	—	—	—	P7 OUT2	P7 OUT1	P7 OUT0
Reset:	0	0	0	0	0	0	0	0

P7OUTn(n=2-0)	Port 7 Output Select
0	Low
1	High

Port 7 output register (P7OUT : x'E00717' R/W)

	7	6	5	4	3	2	1	0
P7IN	—	—	—	—	—	P7 IN2	P7 IN1	P7 IN0
Reset:	0	0	0	0	0	Port	Port	Port

P7INn(n=2-0)	Port 7 Input
0	Low
1	High

Port 7 input register (P7IN : x'E00727' R)

	7	6	5	4	3	2	1	0
P7DIR	—	—	—	—	—	P7 DIR2	P7 DIR1	P7 DIR0
Reset:	0	0	0	0	0	0	0	0

P7DIRn(n=2-0)	Port 7 Input/Output Direction Select
0	Input
1	Output

Port 7 input/output control register (P7DIR : x'E00737' R/W)

	7	6	5	4	3	2	1	0
P7MD	—	—	P7 MD5	P7 MD4	P7 MD3	P7 MD2	P7 MD1	P7 MD0
Reset:	0	0	0	0	0	0	0	0

P7MD1	P7MD0	P70 Input/Output Signal Select
0	0	Port/TM7IO input/SBI3 input
0	1	TM7IO output
1	0	Reserved
1	1	/CS1S

P7MD3	P7MD2	P71 Input/Output Signal Select
0	0	Port/TM8IO input
0	1	TM8IO output
1	0	SBO3 output
1	1	/CS2S

P7MD5	P7MD4	P72 Input/Output Signal Select
0	0	Port/TM9IO input/SBT3 input
0	1	TM9IO output
1	0	SBT3 output
1	1	/CS3S

Note: Do not select timer input when using Port 1 as timer.

Port 7 mode register (P7MD : x'E00748' R/W)

11-2-10 Register of Port 8

	7	6	5	4	3	2	1	0
P8PLU	—	—	—	P8 PLU4	P8 PLU3	P8 PLU2	P8 PLU1	P8 PLU0
Reset:	0	0	0	0	0	0	0	0

P8PLUn(n=4-0)	Port 8 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 8 pull-up register (P8PLU : x'E00708' R/W)

	7	6	5	4	3	2	1	0
P8OUT	—	—	—	P8 OUT4	P8 OUT3	P8 OUT2	P8 OUT1	P8 OUT0
Reset:	0	0	0	0	0	0	0	0

P8OUTn(n=4-0)	Port 8 Output Select
0	Low
1	High

Port 8 output register (P8OUT : x'E00718' R/W)

	7	6	5	4	3	2	1	0
P8IN	—	—	—	P8 IN4	P8 IN3	P8 IN2	P8 IN1	P8 IN0
Reset:	0	0	0	Port	Port	Port	Port	Port

P8INn(n=4-0)	Port 8 Input
0	Low
1	High

Port 8 input register (P8IN : x'E00728' R)

	7	6	5	4	3	2	1	0
P8DIR	—	—	—	P8 DIR4	P8 DIR3	P8 DIR2	P8 DIR1	P8 DIR0
Reset:	0	0	0	0	0	0	0	0

P8DIRn(n=4-0)	Port 8 Input/Output Direction Select
0	Input
1	Output

Port 8 input/output control register (P8DIR : x'E00738' R/W)

	7	6	5	4	3	2	1	0
P8MD	P8 MD7	P8 MD6	P8 MD5	P8 MD4	P8 MD3	P8 MD2	P8 MD1	P8 MD0
Reset:	0	0	0	0	0	0	0	0

P8MD0	P80 Input/Output Signal Select A
0	Port/TM10IOA input/WDOUT output
1	TM10IOA output

P8MD1	P81 Input/Output Signal Select A
0	Port/TM10IOB input/STOP output
1	TM10IOB output

P8MD3	P8MD2	P83 Input/Output Signal Select
0	0	Port/TM11IOA input
0	1	TM11IOA output
1	0	SBO2 output

P8MD5	P8MD4	P84 Input/Output Signal Select
0	0	Port/TM11IOB input/SBT2 input
0	1	TM11IOB output
1	0	SBT2 output

P8MD6	P80 Input/Output Signal Select B
0	Port
1	WDOUT output (Set P8DIR0 to 1)

P8MD7	P81 Input/Output Signal Select B
0	Port
1	STOP output (Set P8DIR1 to 1)

Port 8 mode register (P8MD : x'E00749' R/W)

P82 pin is used as SBI2 pin as well, but because the SBI2 pin is only for input, so when reset is released, P82 pin can be used in SBI2 input mode by setting to receive serial 2.

11-2-11 Register of Port 9

	7	6	5	4	3	2	1	0
P9PLU	—	—	—	—	P9 PLU3	P9 PLU2	P9 PLU1	P9 PLU0
Reset:	0	0	0	0	0	0	0	0

P9PLUn(n=3-0)	Port 9 Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port 9 pull-up register (P9PLU : x'E00709' R/W)

	7	6	5	4	3	2	1	0
P9OUT	—	—	—	—	P9 OUT3	P9 OUT2	P9 OUT1	P9 OUT0
Reset:	0	0	0	0	0	0	0	0

P9OUTn(n=3-0)	Port 9 Output Select
0	Low
1	High

Port 9 output register (P9OUT : x'E00719' R/W)

	7	6	5	4	3	2	1	0
P9IN	—	—	—	—	P9 IN3	P9 IN2	P9 IN1	P9 IN0
Reset:	0	0	0	0	Port	Port	Port	Port

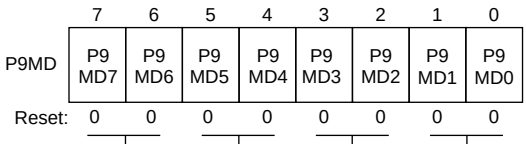
P9INn(n=3-0)	Port 9 Input
0	Low
1	High

Port 9 input register (P9IN : x'E00729' R)

	7	6	5	4	3	2	1	0
P9DIR	—	—	—	—	P9 DIR3	P9 DIR2	P9 DIR1	P9 DIR0
Reset:	0	0	0	0	0	0	0	0

P9DIRn(n=3-0)	Port 9 Input/Output Direction Select
0	Input
1	Output

Port 9 input/output control register (P9DIR : x'E00739' R/W)



P9MD1	P9MD0	P90 Input/Output Signal Select
0	0	Port/TM12IOA input
0	1	TM12IOA output
1	0	AN0 input

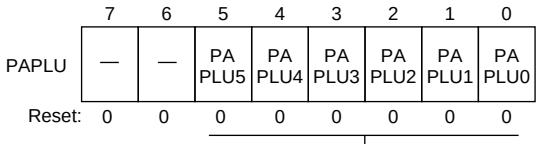
P9MD3	P9MD2	P91 Input/Output Signal Select
0	0	Port/TM12IOB input
0	1	TM12IOB output
1	0	AN1 input

P9MD5	P9MD4	P92 Input/Output Signal Select
0	0	Port/TM13IOA input
0	1	TM13IOA output
1	0	AN2 input

P9MD7	P9MD6	P93 Input/Output Signal Select
0	0	Port/TM13IOB input
0	1	TM13IOB output
1	0	AN3 input

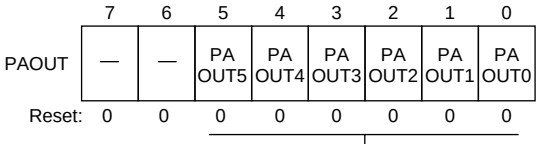
Port 9 mode register (P9MD : x'E0074A' R/W)

11-2-12 Register of Port A



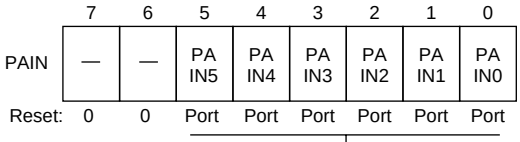
PAPLUn(n=5-0)	Port A Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port A pull-up register (PAPLU : x'E0070A' R/W)



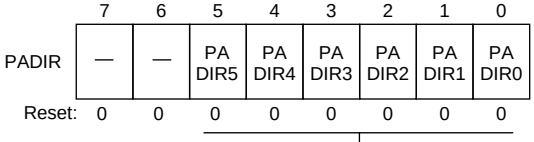
PAOUTn(n=5-0)	Port A Output Select
0	Low
1	High

Port A output register (PAOUT : x'E0071A' R/W)



PAINn(n=5-0)	Port A Input
0	Low
1	High

Port A input register (PAIN : x'E0072A' R)



PADIRn(n=5-0)	Port A Input/Output Direction Select
0	Input
1	Output

Port A input/output control register (PADIR : x'E0073A' R/W)

	7	6	5	4	3	2	1	0
PALMD	—	PA LMD6	PA LMD5	PA LMD4	PA LMD3	PA LMD2	PA LMD1	PA LMD0
Reset:	0	0	0	0	0	0	0	0

PALMD0	PA0 Input/Output Signal Select
0	Port/SBI1 input
1	AN4 input

PALMD2	PALMD1	PA1 Input/Output Signal Select
0	0	Port
0	1	SBO1 output
1	0	SDA1 open-drain input/output
1	1	AN5 input

PALMD5	PALMD4	PALMD3	PA2 Input/Output Signal Select
0	0	0	Port/SBT1 input
0	0	1	SBT1 output
0	1	0	SBT1 half-duplex output
0	1	1	SCL1 open-drain input/output
1	0	0	AN6 input

PALMD6	PA3 Input/Output Signal Select
0	Port/SBI0 input
1	AN7 input

Port A mode register L (PALMD : x'E0074C' R/W)

	7	6	5	4	3	2	1	0
PAHMD	—	—	—	—	PA HMD3	PA HMD2	PA HMD1	PA HMD0
Reset:	0	0	0	0	0	0	0	0

PAHMD1	PAHMD0	PA4 Input/Output Signal Select
0	0	Port
0	1	SBO0 output
1	0	SDA0 open-drain input/output

PAHMD3	PAHMD2	PA5 Input/Output Signal Select
0	0	Port/SBT0
0	1	SBT0 output
1	0	SBT0 half-duplex output
1	1	SCL0 open-drain input/output

Port A mode register H (PAHMD : x'E0074D' R/W)

11-2-13 Register of Port B

	7	6	5	4	3	2	1	0
PBPLU	—	—	PB PLU5	PB PLU4	PB PLU3	PB PLU2	PB PLU1	PB PLU0
Reset:	0	0	0	0	0	0	0	0

PBPLUn(n=5-0)	Port B Pullup Select
0	Pullup resistor off
1	Pullup resistor on

Port B pull-up register (PBPLU : x'E0070B' R/W)

	7	6	5	4	3	2	1	0
PBOUT	—	—	PB OUT5	PB OUT4	PB OUT3	PB OUT2	PB OUT1	PB OUT0
Reset:	0	0	0	0	0	0	0	0

PBOUTn(n=5-0)	Port B Output Select
0	Low
1	High

Port B output register (PBOUT : x'E0071B' R/W)

	7	6	5	4	3	2	1	0
PBIN	—	NNMI	PB IN5	PB IN4	PB IN3	PB IN2	PB IN1	PB IN0
Reset:	0	Port	Port	Port	Port	Port	Port	Port

PBINn(n=5-0)	Port B Input
0	Low
1	High

Port B input register (PBIN : x'E0072B' R)

	7	6	5	4	3	2	1	0
PBDIR	—	—	PB DIR5	PB DIR4	PB DIR3	PB DIR2	PB DIR1	PB DIR0
Reset:	0	0	0	0	0	0	0	0

PBDIRn(n=5-0)	Port B Input/Output Direction Select
0	Input
1	Output

Port B input/output control register (PBDIR : x'E0073B' R/W)

PB0 to PB5 pins are used as /IRQ0 to /IRQ5 signals as well. They can be used by permitting the external pin interrupt by the maskable interrupt control register (G1ICR, G2ICR, G3ICR).

11-3 Port Block Diagram

The MN102H74G/74F/74D/F74G has twelve ports of P0 to PB. Each port contains 3 pins to 8 pins. Each pin has a port function and an input/output control function of each peripheral. The function is selected by setting the each port mode register. The port input/output direction is determined by setting each port mode register when the peripheral function is selected as an input/output control pin of the peripheral. The port input/output direction is determined by setting each port direction control register when the port is selected as a general-purpose port. Each port has a pullup resistor controlled by software and switched on or off regardless of the port mode register setup or the port direction control register setup.

Table 11-3-1 Port Block Diagram (1/11)

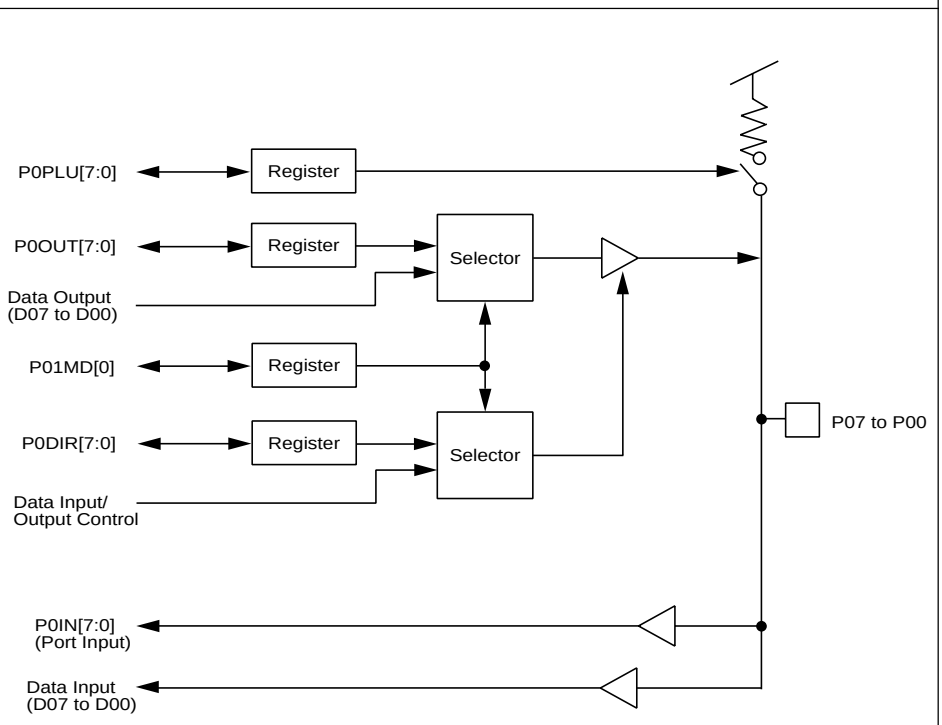
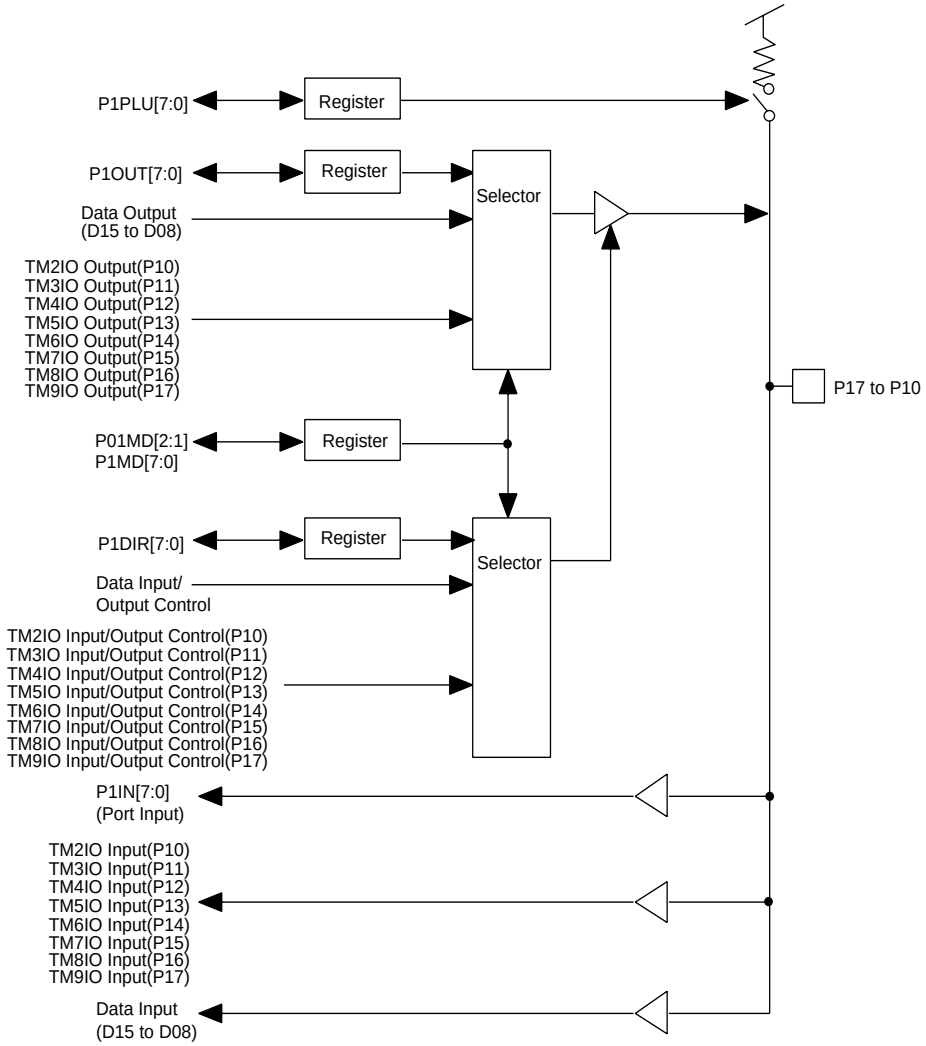
Port	Associated Pins (Shared Pins)	Block Diagram
Port 0	P07 to P00 D07 to D00	 Note: The P0DIR register value is valid only when Port 0 and Port 1 are selected as port in the P01MD register.

Table 11-3-1 Port Block Diagram (2/11)

Port	Associated Pins (Shared Pins)	Block Diagram
Port 1	P17 to P10 D15 to D08 TM9IO to TM2IO	

Note: The P1DIR register value is valid only when port is selected in the P01MD register and the P1MD register.
The TMnIO input/output direction is determined automatically by setting the P01MD register and P1MD register.
Port 4 and port 7 cannot be used as timer input when port 1 is selected as timer.

Table 11-3-1 Port Block Diagram (3/11)

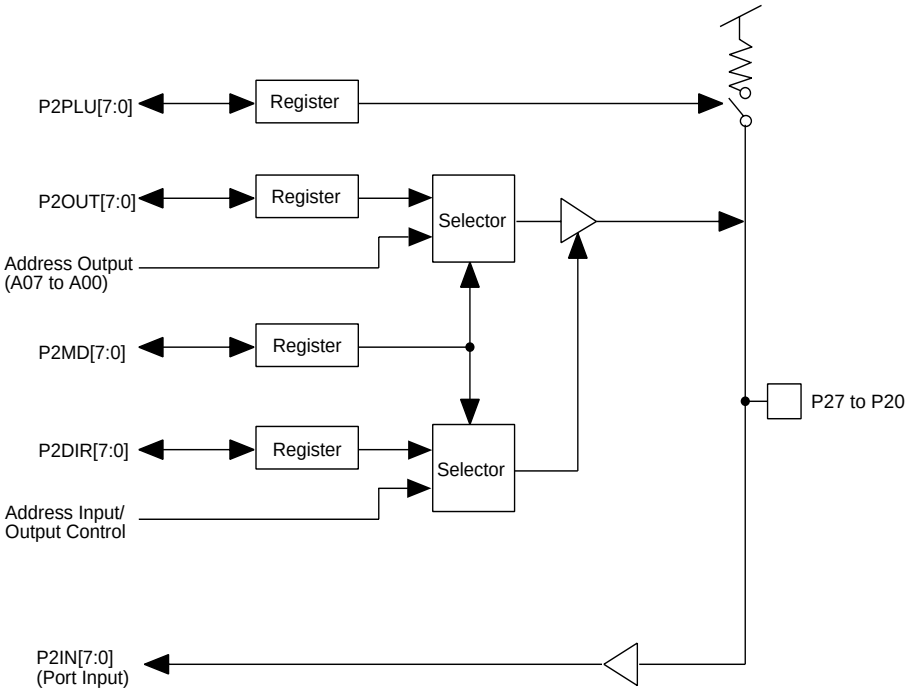
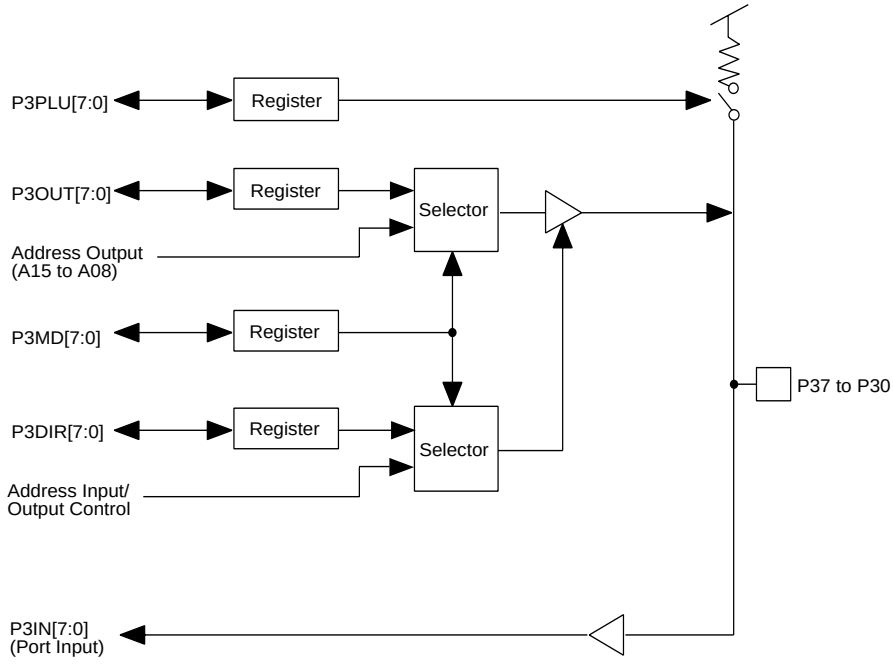
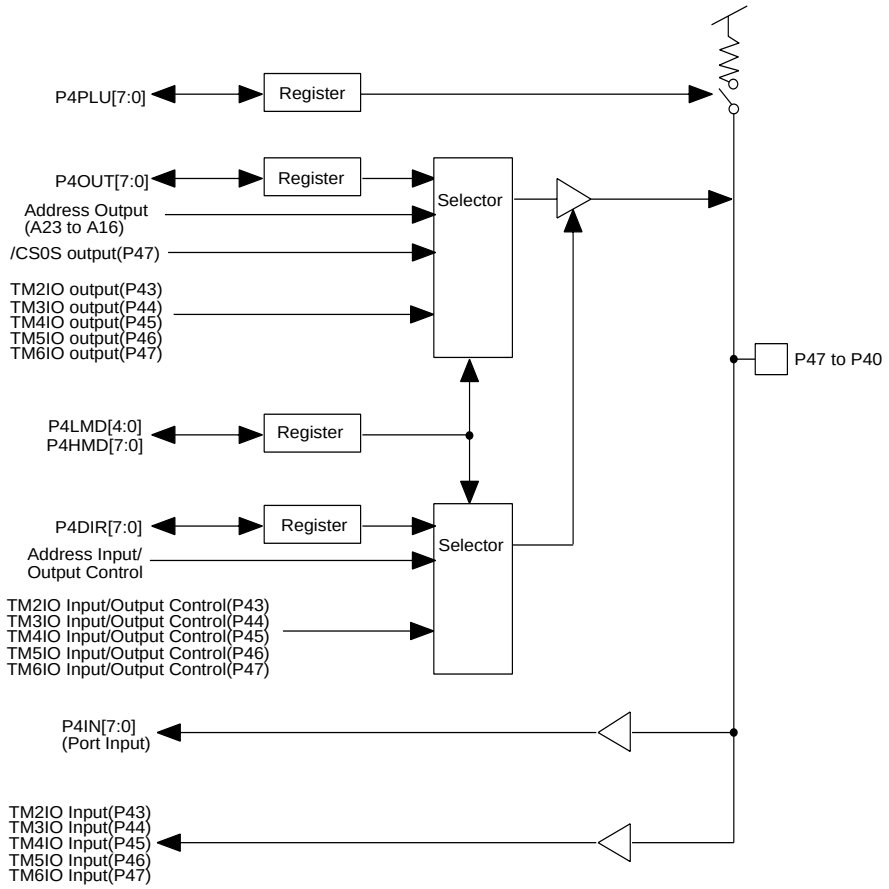
Port	Associated Pins (Shared Pins)	Block Diagram
Port 2	P27 to P20 A07 to A00	 Note: The P2DIR register value is valid only when port is selected in the P2MD register.
Port 3	P37 to P30 A15 to A08	 Note: The P3DIR register value is valid only when port is selected in the P3MD register.

Table 11-3-1 Port Block Diagram (4/11)

Port	Associated Pins (Shared Pins)	Block Diagram
Port 4	P47 to P40 A23 to A16 TM6IO to TM2IO	

Note: The P4DIR register value is valid only when port is selected in the P4LMD register and the P4HMD register.
The TMnIO input/output direction is determined automatically by setting the P4LMD register and the P4HMD register.
Do not use port 4 as timer input when using port 1 as timer.

Table 11-3-1 Port Block Diagram (5/11)

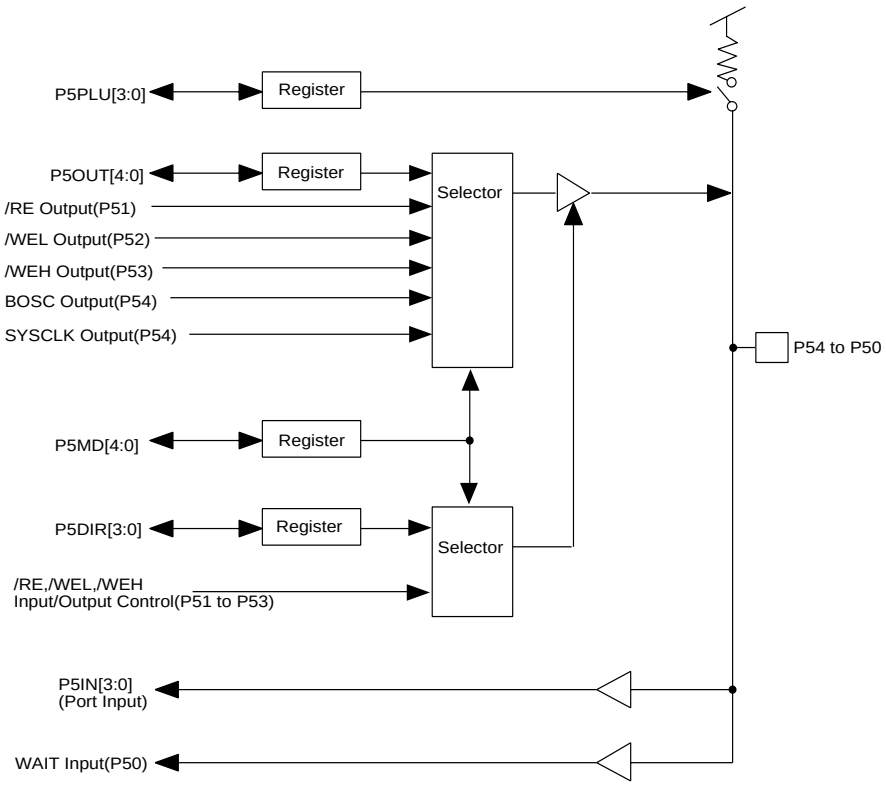
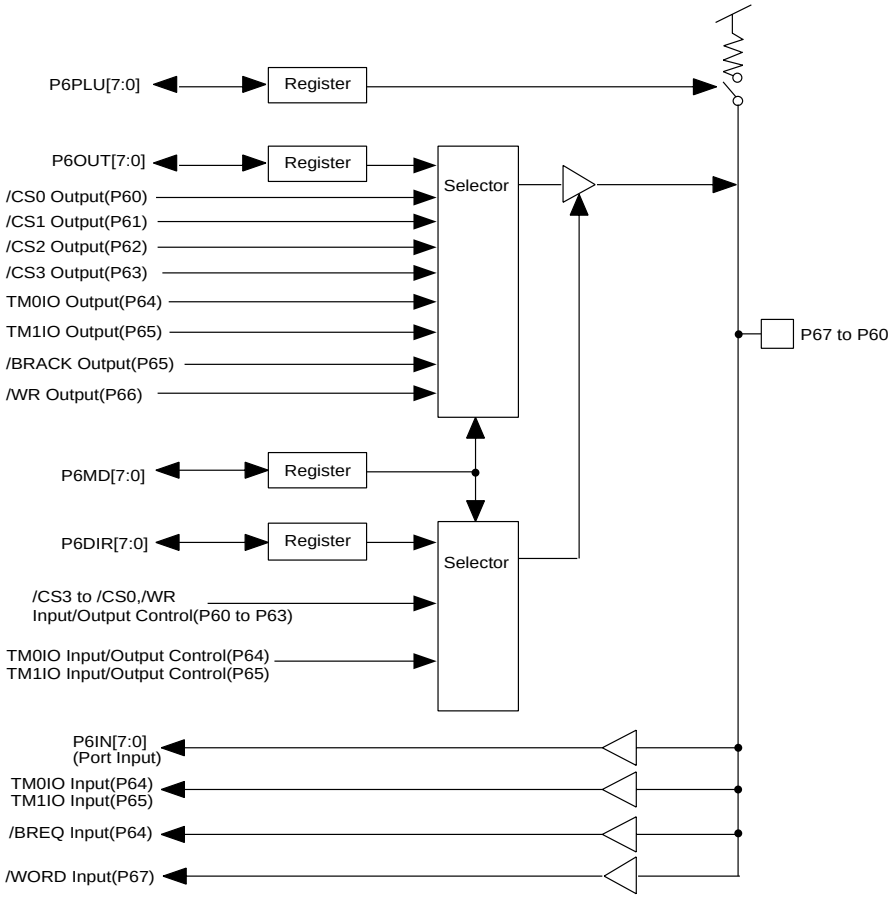
Port	Associated Pins (Shared Pins)	Block Diagram
Port 5	P54 to P50 WAIT /RE /WEL /WEH BOSC SYSCLK	 Note: The P5DIR register value is valid only when port is selected in the P5MD register.

Table 11-3-1 Port Block Diagram (6/11)

Port	Associated Pins (Shared Pins)	Block Diagram
Port 6	P67 to P60 /CS3 to /CS0 /BREQ /BRACK /WR /WORD TM0IO TM1IO	

Note: The P6DIR register value is valid only when port is selected in the P6MD register.
The /CS3 to /CS0, /WR, /BREQ, /BRACK, and TMnIO input/output direction is determined automatically by setting the P6MD register.
Use P67 as a port only when single-chip mode is selected.

Table 11-3-1 Port Block Diagram (7/11)

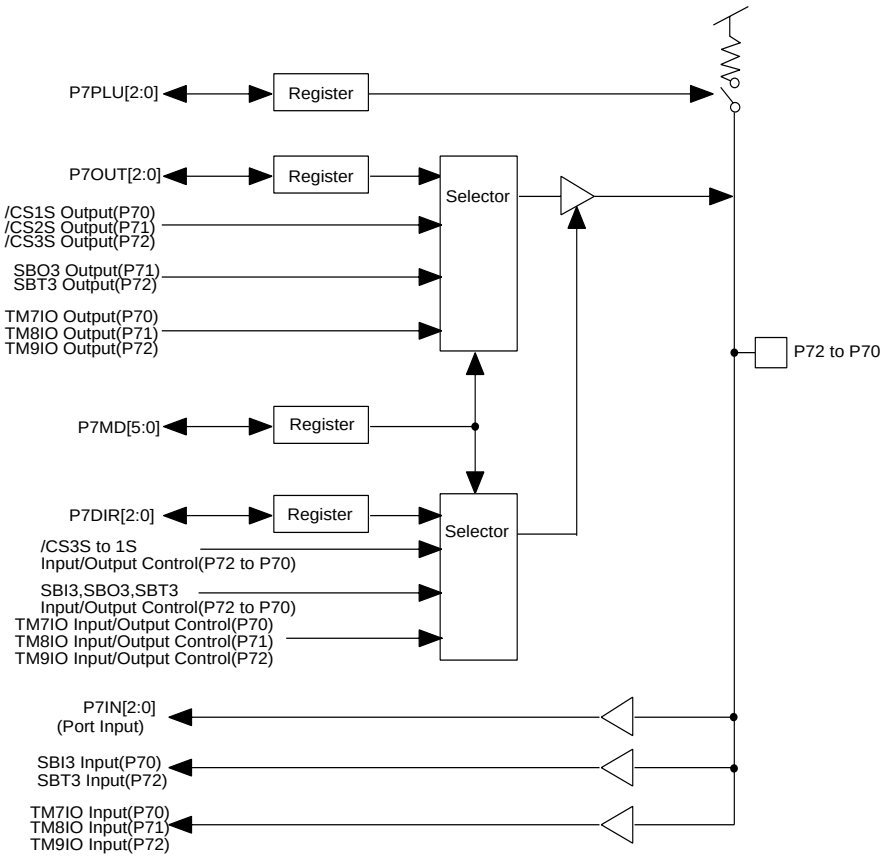
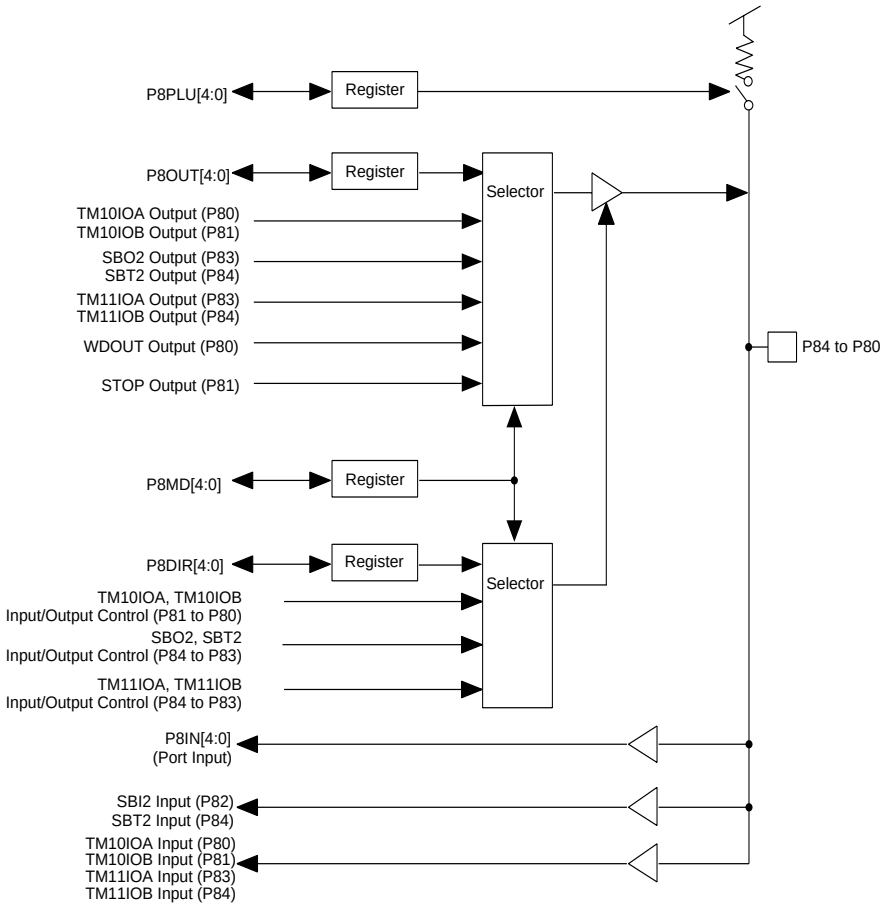
Port	Associated Pins (Shared Pins)	Block Diagram
Port 7	P72 to P70 TM9IO to TM7IO SBI3 SBO3 SBT3 /CS3S to /CS1S	 Note: The P7DIR register value is valid only when port is selected in the P7MD register. The TMnIO, SBI3, SBO3, SBT3, /CS1S to /CS3S input/output direction is determined automatically by setting the P7MD register. Do not use port 7 as timer input when Port 1 is used as timer.

Table 11-3-1 Port Block Diagram (8/11)

Port	Associated Pins (Shared Pins)	Block Diagram
Port 8	P84 to P80 TM10IOA TM10IOB TM11IOA TM11IOB SBI2 SBO2 SBT2 WDOUT STOP	

Note: The P8DIR register value is valid only when port and WDOUT output and STOP output is selected in the P8MD register.
The TMnIOA, TMnIOB, SBI2, SBO2, SBT2 input/output direction is determined automatically by setting the P8MD register.

Table 11-3-1 Port Block Diagram (9/11)

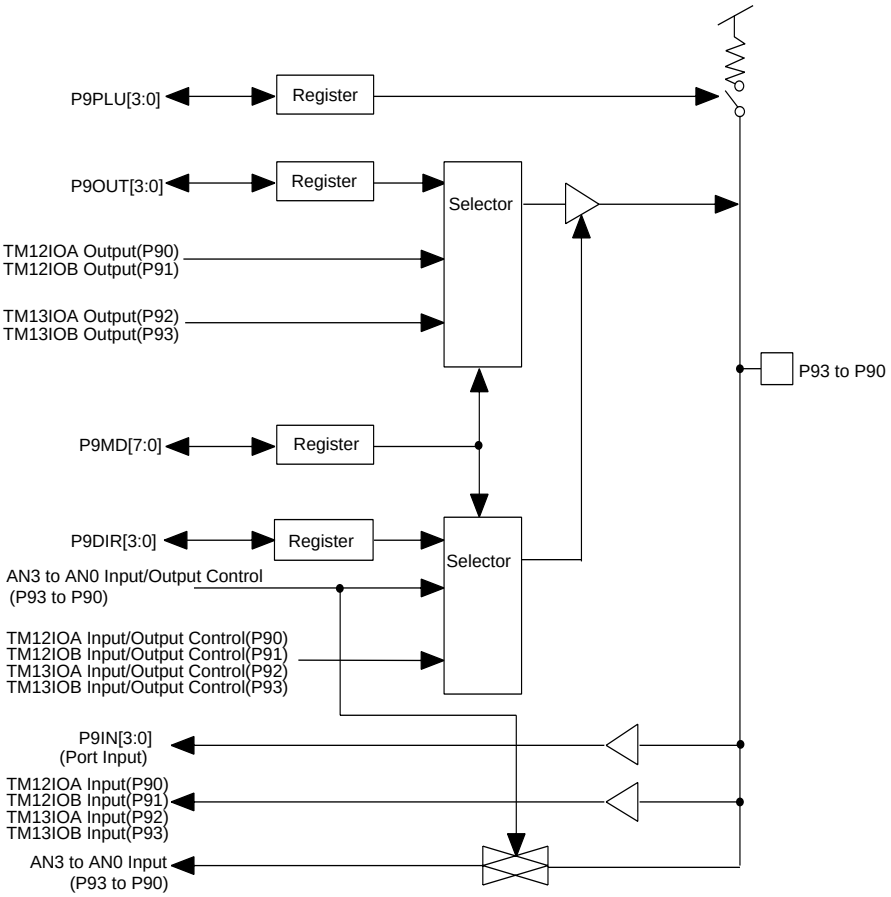
Port	Associated Pins (Shared Pins)	Block Diagram
Port 9	P93 to P90 AN3 to AN0 TM12IOA TM12IOB TM13IOA TM13IOB	 Note: The P9DIR register value is valid only when port is selected in the P9MD register. The TMnIOA, TMnIOB, AN3 to AN0 input/output direction is determined automatically by setting the P9MD register.

Table 11-3-1 Port Block Diagram (10/11)

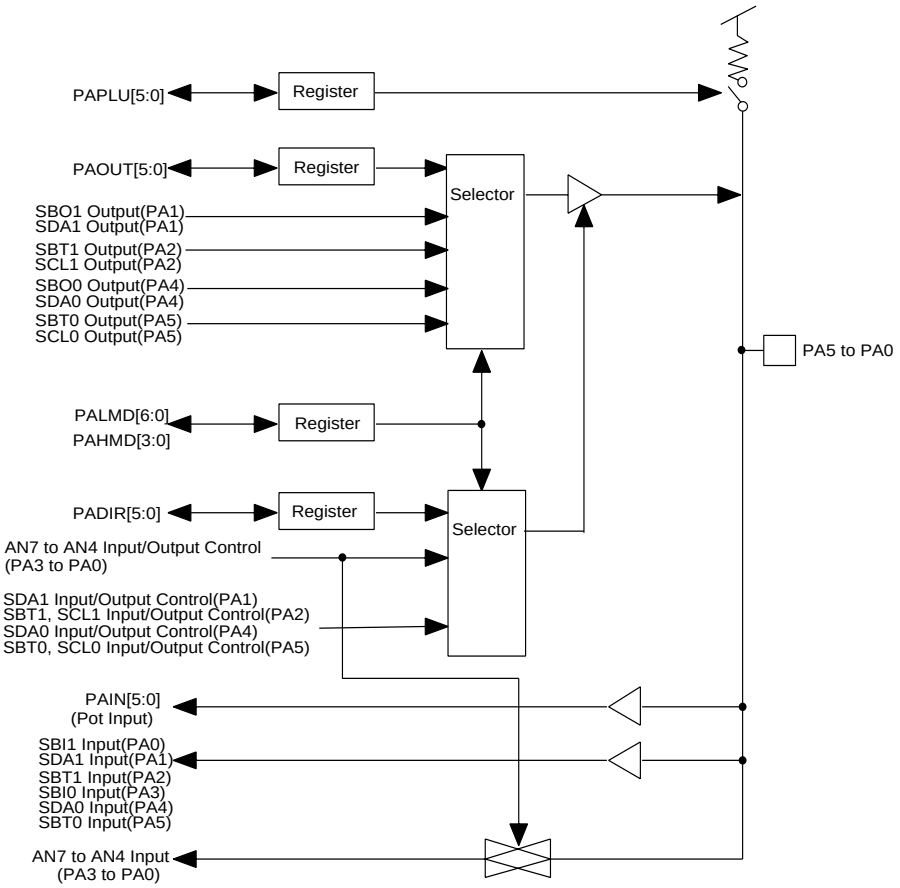
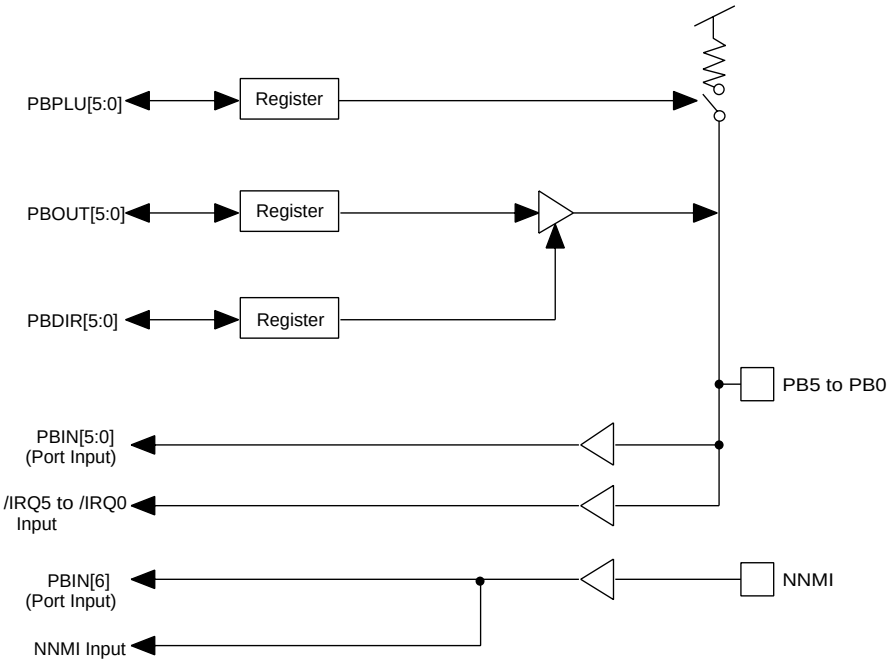
Port	Associated Pins (Shared Pins)	Block Diagram
Port A	PA5 to PA0 AN7 to AN4 SBI0 SBO0 SBT0 SBI1 SBO1 SBT1 SDA0 SCL0 SDA1 SCL1	 Note: The PADIR register value is valid only when port is selected in the PALMD and PAHMD register. The SDAn, SCLn, SBTn, AN7 to AN4 input/output direction is determined automatically by setting the PALMD and PAHMD register.

Table 11-3-1 Port Block Diagram (11/11)

Port	Associated Pins (Shared Pins)	Block Diagram
Port B	PB5 to PB0 /IRQ5 to /IRQ0 /NMI	 Note: When /IRQn signal is used, set up the bit of PBDIR register to 0 (input, default value).

11-4 Port Setup Examples

11-4-1 General-purpose Port Setup

This section describes a light-emitting diode (LED) on/off based on switch input status. P71 is connected to the switch and P70 is connected to the LED. In this configuration, the LED is on when the switch is on while the LED is off when the switch is off.

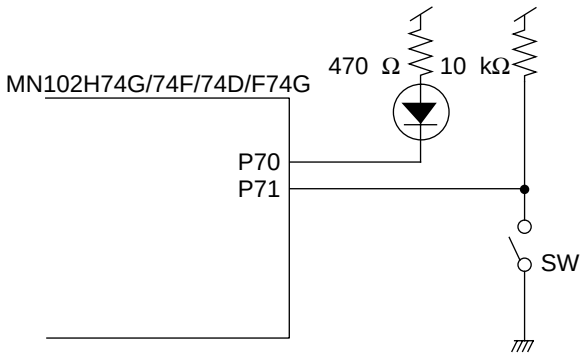


Figure 11-4-1 General-purpose Port Setup Example

1) Both P71 pin and P70 pin are set to the initial values after reset release. Under this condition, the LED is off. Next, set the P70 pin to the general-purpose port output.

P7DIR : x'E00737'

7	6	5	4	3	2	1	0
-	-	-	-	-	P7 DIR2	P7 DIR1	P7 DIR0
-	-	-	-	-	0	0	1

P7MD : x'E00748'

7	6	5	4	3	2	1	0
-	-	P7 MD5	P7 MD4	P7 MD3	P7 MD2	P7 MD1	P7 MD0
-	-	0	0	0	0	0	0

2) Read the P71 pin status (P7IN) with the MOVB instruction. If bit 1 is '0' (switch on), set P7OUT to x'00'.

P7OUT : x'E00717'

7	6	5	4	3	2	1	0
-	-	-	-	-	P7 OUT2	P7 OUT1	P7 OUT0
-	-	-	-	-	0	0	0

P7IN : x'E00727'

7	6	5	4	3	2	1	0
-	-	-	-	-	P7 IN2	P7 IN1	P7 IN0

If bit 1 is ‘1’ (switch off), set P7OUT to x’01’.

P7OUT : x’E00717’

7	6	5	4	3	2	1	0
-	-	-	-	-	P7 OUT2	P7 OUT1	P7 OUT0
-	-	-	-	-	0	0	1

Under this condition, the ‘L’ level is output to P70 if the switch is on while the ‘H’ level is output to P70 if the switch is off resulting that the LED is on/off. Thereafter, reading the P71 pin status is repeated.

Figure 11-4-2 and 11-4-3 show the flowcharts of general-purpose port operations. When the port is input, set the PnMDm flag and PnDIRn flag (n = port number, m = bit position) to ‘0’ and read the InINm flag. When the port is output, set PnDIRm flag to ‘1’ and write the data output to the PnOUTm flag. Regardless of input/output direction, set the PnPLUm flag to ‘1’ for the pull-up setting.

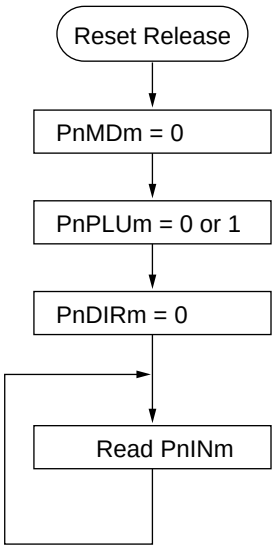


Figure 11-4-2 Basic Flowchart of Genera-purposel Port Input

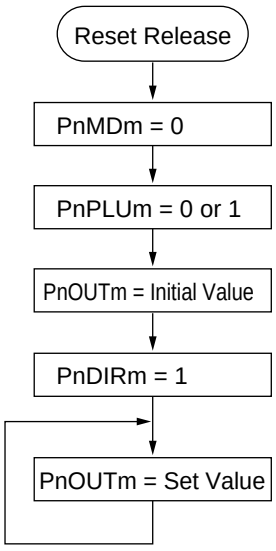


Figure 11-4-3 Basic Flowchart of Genera-purpose Port Output

Chapter 12 Appendix

12-1 Electrical Characteristics

Structure	CMOS integrated circuit
Application	General purpose
Function	16-bit microcontroller
Pin Configuration	Figure 1-4-1 to Figure 1-4-3
External Dimension	Figure 1-4-7

A. Absolute Maximum Ratings

V_{SS} = 0 V

Parameter		Symbol	Rating	Unit
A1	Power supply voltage	V_{DD}	-0.3 to +4.6	V
A2	Input pin voltage	V_I	-0.3 to V_{DD}+0.3	V
A3	Output pin voltage	V_O	-0.3 to V_{DD}+0.3	V
A4	Input/output pin voltage	V_{IO}	-0.3 to V_{DD}+0.3	V
A5	Operating ambient temperature	Topr	-20 to +70	°C
A6	Storage temperature	Tstg	-55 to +125	°C

Note:

(1) Absolute maximum ratings are stress ratings not to cause damage to the device, and these ratings are not ratings to guarantee operations.

(2) All of V_{DD} and V_{SS} pins are external pins. Connect them directly to the power source and ground.

(3) To prevent latch-up tolerance, connect more than one bypass condenser between V_{DD} pin and V_{SS} pin. The condenser must be inserted close to those pins. Use at least 0.2 μF condenser.

This LSI User's manual describes standard specifications. Contact one of our sales offices for product standards when using this LSI chip.

B. Operating Conditions

V_{SS} = 0 V
Ta = -20 °C to +70 °C

	Parameter	Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
B1	Power supply voltage	V_{DD}		3.0	3.3	3.6	V
	Crystal Oscillator 1 (OSCI)						
B2	Oscillator frequency	Fosc1		11.976	12.000	12.024	MHz
	Crystal Oscillator 2 (XI)						
B3	Oscillator frequency	Fosc2		32		166	kHz

C. Electrical Characteristics

(1) DC Characteristics

V_{DD} = 3.3 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

Parameter		Symbol	Conditions		Capacitance			Unit
					Min	Typ	Max	
C1	Power supply current during operation	$I_{DD}1$	$V_I = V_{DD}$ or V_{SS} Fosc1 = 12 MHz output pins are open				65+10α^*	mA
C2	Power supply current during SLOW mode	$I_{DD}2$	$V_I = V_{DD}$ or V_{SS} Fosc2 = 32 kHz output pins are open				5	mA
C3	Power supply current during STOP mode	$I_{DD}3$	stop oscillation stop all functions	Ta = 25 °C		1		µA
				Ta = 70 °C			70	µA
C4	Power supply current during HALT 0 mode	$I_{DD}4$	Fosc1 = 12 MHz Fosc2 = 32 kHz				30+10α^*	mA
C5	Power supply current during HALT 1 mode	$I_{DD}5$	Fosc1 = stop oscillation Fosc2 = 32 kHz				2	mA

*α depends on the model.

MN102H74G	α = 0
MN102H74F	α = 0
MN102H74D	α = 0
MN102HF74G	α = 1

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Input/Output pin 1 < Output push-pull / Input LVTTL level schmidt trigger / Programmable pull-up >: P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P53, P60 to P67, P70 to P72, P80 to P84, PA4 to PA5, PB0 to PB5							
C6	Input high voltage	V_{IH1}		2.2			V
C7	Input low voltage	V_{IL1}				0.6	V
C8	Output high voltage	V_{OH1}	I_{OH} = -2.0 mA V_{DD} = 3.3 V	2.4			V
C9	Output low voltage	V_{OL1}	I_{OL} = 2.0 mA V_{DD} = 3.3 V			0.4	V
C10	Output leakage current	I_{LO1}	Vo = Hi-z V_I = V_{DD} or V_{SS}	-10		10	μA
C11	Pull-up resistor	P_{PU1}	V_I = V_{SS} V_{DD} = 3.3 V	10	30	90	kΩ
Input/Output pin 2 < Output push-pull / Input LVTTL level schmidt trigger / Programmable pull-up / Analog pin > : P90 to P93, PA0 to PA3							
C12	Input high voltage	V_{IH2}		2.2			V
C13	Input low voltage	V_{IL2}				0.6	V
C14	Output high voltage	V_{OH2}	I_{OH}=-2.0 mA V_{DD}=3.3 V	2.4			V
C15	Output low voltage	V_{OL2}	I_{OL}=2.0 mA V_{DD}=3.3 V			0.4	V
C16	Output leakage current	I_{LO2}	Vo=Hi-z VI=V_{DD} or V_{SS}	-10		10	μA
C17	Pull-up resistor	P_{PU2}	V_I=V_{SS} V_{DD}=3.3 V	10	30	90	kΩ

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Input/Output pin 3 < USB port > : D+, D-							
C18	Output high voltage	V_{OH3}	15 kΩ to Vss (Note)	2.8			V
C19	Output low voltage	V_{OL3}	1.5 kΩ to 3.6 V (Note)			0.3	V
C20	Output leakage current	I_{LO3}	Vo = Hi-z $V_I = V_{DD}$ or V_{SS}	-10		10	μA
C21	Output Impedance	Z_{DRV}		4		20	Ω
Output pin < Output push-pull > : P54							
C22	Output high voltage	V_{OH4}	I_{OH} = -2.0 mA V_{DD} = 3.3 V	2.4			V
C23	Output low voltage	V_{OL4}	I_{OL} = 2.0 mA V_{DD} = 3.3 V			0.4	V

Note: Connect resistor of 24 Ω in series in external.

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Input pin < Input CMOS level schmidt trigger > : /NMI, MODE, /RST, USBMODE							
C24	Input high voltage	V_{IH4}		V_{DD}×0.9			V
C25	Input hlow voltage	V_{IL4}				V_{DD}×0.1	V
C26	Input leakage current	I_{LO5}	V_{DD} = 3.6 V VI = V_{DD} or V_{SS}	-10		10	μA
OSCI, XI pin (at input external clock) : See Figure 1-4-4, 1-4-5 for self-excited oscillation of crystal or ceramic oscillator.							
C27	Input high voltage	V_{IH5}		V_{DD}×0.8		V_{DD}	V
C28	Input low voltage	V_{IL5}		V_{SS}		V_{DD}×0.2	V
pin capacity							
C29	Input pin	C_{IN}	Ta = 25 °C		7		pF
C30	Output pin	C_{OUT}			7		pF
C31	Input/output pin	$C_{I/O}$			7		pF

D. A/D Converter Characteristics

V_{DD} = 3.3 V
V_{SS} = 0 V
Ta = 25 °C

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
D1	Resolution	L_{SB1}				10	Bits
D2	Zero-scale transition voltage	V_{ZS}	V_{DD} = 3.3 V V_{SS} = 0 V	-20		20	mV
D3	Full-scale transition voltage	V_{FS1}	V_{DD} = 3.3 V V_{SS} = 0 V	3.25		3.35	V
D4	Non-linearity error	N_{LE1}	V_{DD} = 3.3 V V_{SS} = 0 V	-4		4	LSB
D5	Differential non-linearity error	D_{NLE1}	V_{DD} = 3.3 V V_{SS} = 0 V	-4		4	LSB
D6	A/D conversion time	T_{SET1}	Fosc = 12 MHz	2.00			µs
D7	A/D conversion cycle	T_{SET2}	Fosc = 12 MHz	2.00			µs
D8	Analog input voltage	V_{IA}		V_{SS}		V_{DD}	V

E. AC Characterisitics (input)

Input Timing Conditions

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
External clock input timing (Fosc1 = 12 MHz)							
E1	External clock input cycle time	t_{EXCcyc}	Figure 12-1-1	83.2	83.3	83.5	ns
E2	External clock input high pulse width	t_{EXCH}		$\frac{t_{EXCcyc}}{2}$ -5			ns
E3	External clock input low pulse width	t_{EXCL}		$\frac{t_{EXCcyc}}{2}$ -5			ns
E4	External clock input rise time	t_{EXCR}				5	ns
E5	External clock input fall time	t_{EXCF}				5	ns
Reset input timing							
E6	Reset signal pulse width (/RST)	t_{RSTW}	Figure 12-1-2	2			t_{EXCcyc}
Power rise timing							
E7	V_{DD}-/RST setup time	t_{RSTS}	Figure 12-1-3	2			ms

Input Timing Conditions

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

	Parameter	Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Data transfer signal input timing							
E8	Data acknowledge signal setup time 1 (WAIT)	t_{WS1}	Figure12-1-5	25			ns
E9	Data acknowledge signal hold time 1 (WAIT)	t_{WH1}		0			ns
E10	Data acknowledge signal setup time 2 (WAIT)	t_{WS2}	Figure 12-1-7	25			ns
E11	Data acknowledge signal hold time 2 (WAIT)	t_{WH2}		0			ns
Data transfer signal input timing							
E12	Read data setup time (D15-00)	t_{RDS}	Figure 12-1-4 Figure 12-1-5	25			ns
E13	Read data hold time 1 (D15-00)	t_{RDH1}		0			ns
E14	Read data hold time 2 (D15-00)	t_{RDH2}	Figure 12-7	0			ns
Bus open request signal input timing							
E15	Bus open request signal setup time (/BREQ)	t_{BREQS}	Figure 12-1-9	0			ns
E16	Bus open request signal hold time (/BREQ)	t_{BREQH}		0			ns
Interrupt signal input timing							
E17	Non-maskable Interrupt signal pulse width (/NMI)	t_{NMIW}	Figure 12-1-10	10 (Note)			t_{cyc}
E18	External Interrupt signal pulse width (/IRQ5 to /IRQ0)	t_{IRQW}		4 (Note)			t_{cyc}

Note: Interrupt can be generated when the noise under the time is input.

Input Timing Conditions

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C

	Parameter	Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Serial interface signal timing (synchronous serial reception)							
E19	Data reception setup time (SBI3-0)	t_{RXDS}	Figure 12-1-13	60			ns
E20	Data reception hold time (SBI3-0)	t_{RXDH}		60			ns
E21	Transfer clock input high pulse width (SBT3-0)	t_{SCH}	Figure 12-1-13	t_{cyc}×2			ns
E22	Transfer clock input low pulse width (SBT3-0)	t_{SCL}		t_{cyc}×2			ns
Timer counter signal input timing							
E23	Timer External clock low pulse width (TMnIO:n = 0-9) (TMnIOA, TMnIOB:n = 10-13)	t_{TCCLKL}	Figure 12-1-14	t_{cyc}×2			ns
E24	Timer External clock high pulse width (TMnIO:n = 0-9) (TMnIOA, TMnIOB:n = 10-13)	t_{TCCLKH}		t_{cyc}×2			ns

F. AC Characterisitics (Output)

Output Signal Characterisitics

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C
C_L = 70 pF

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
System Clock (BOSC/SYSCLK) output timing							
F1	Sytem Clock output cycle time 1 (BOSC)	t_{CYC}	Figure 12-1-1		41.6		ns
F2	Sytem Clock output low pulse width 1 (BOSC)	t_{CL}		15.8			ns
F3	Sytem Clock output high pulse width 1 (BOSC)	t_{CH}		15.8			ns
F4	Sytem Clock output rise time 1 (BOSC)	t_{CR}				5	ns
F5	Sytem Clock output fall time 1 (BOSC)	t_{CF}				5	ns
F6	Sytem Clock output cycle time 2 (SYSCLK)	t_{SYSCYC}	Figure 12-1-1		83.3		ns
F7	Sytem Clock output low pulse width 2 (SYSCLK)	t_{SYSL}		36.6			ns
F8	Sytem Clock output high pulse width 2 (SYSCLK)	t_{SYSH}		36.6			ns
F9	Sytem Clock output rise time 2 (SYSCLK)	t_{SYSR}				5	ns
F10	Sytem Clock output fall time 2 (SYSCLK)	t_{SYSF}				5	ns

Output Signal Characteristics

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C
C_L = 70 pF

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Data transfer signal output timing 1							
F11	Adress delay time (A23-0)	t_{AD}	Figure 12-1-4 to 7			30	ns
F12	Adress hold time (A23-0)	t_{AH}	Figure 12-1-4 to 6	$\frac{tcyc}{2}$ -15			ns
F13	Write data delay time (D15-0)	t_{WDD}	Figure12-1-4 to 7			30	ns
F14	Write data hold time 1 (D15-0)	t_{WDH1}	Figure 12-1-4 to 6	$\frac{tcyc}{2}$ -15			ns
F15	Write data hold time 2 (D15-0)	t_{WDH2}	Figure 12-1-7	tcyc -15			ns

Note: Set the external wait cycle to 0.5 or larger.

Output Signal Characterisitics

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C
C_L = 70 pF

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Data transfer signal output timing 2							
F16	Chip select signal fall delay time (/CS3-0, /CS3S-0S)	t_{CSDF}	Figure 12-1-4 to 7			35	ns
F17	Chip select signal rise delay time (/CS3-0, /CS3S-0S)	t_{CSDR}				35	ns
F18	Chip select signal hold time (/CS3-0, /CS3S-0S)	t_{CSH}	Figure 12-1-4 to 6	$\frac{tcyc}{2}$ -15			ns

Note: Set the external wait cycle to 0.5 or larger.

Output Signal Characteristics

V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C
C_L = 70 pF

	Parameter	Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Data transfer signal output timing 3							
F19	Read enable signal fall delay time (/RE)	t_{REDF}	Figure 12-1-4, 5, 7			25	ns
F20	Read enable signal rise delay time (/RE)	t_{REDR}	Figure 12-1-4, 5, 7			25	ns
F21	Read enable signal setup time (/RE)	t_{SYSRES}	Figure 12-1-8	tcyc-15			ns
F22	Read enable signal hold time (/RE)	t_{SYSREH}	Figure 12-1-8	tcyc-15			ns
F23	Write enable signal fall delay time (/WEH, /WEL)	t_{WEDF}	Figure 12-1-4 to 7			25	ns
F24	Write enable signal rise delay time (/WEH, /WEL)	t_{WEDR}	Figure 12-1-4 to 7			25	ns
F25	Write enable signal pulse width (/WEH, /WEL)	t_{WEPW}	Figure 12-1-4 to 6	30			ns
F26	External access direction signal fall delay time (/WR)	t_{WRDF}	Figure 12-1-6			25	ns
F27	External access direction signal rise delay time (/WR)	t_{WRDR}	Figure 12-1-6			35	ns
F28	External access direction signal pulse width (/WR)	t_{WRPW}	Figure 12-1-6	30			ns
F29	Write enable signal setup time (/WR, /WEH, /WEL)	t_{SYSWES}	Figure 12-1-8	tcyc-15			ns
F30	Write enable signal hold time (/WR, /WEH, /WEL)	t_{SYSWEH}	Figure 12-1-8	tcyc-15			ns

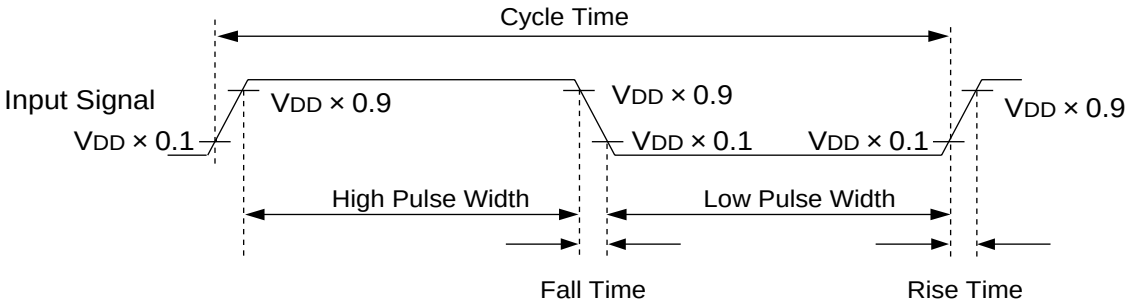
Note: Set the external wait cycle to 0.5 or larger.

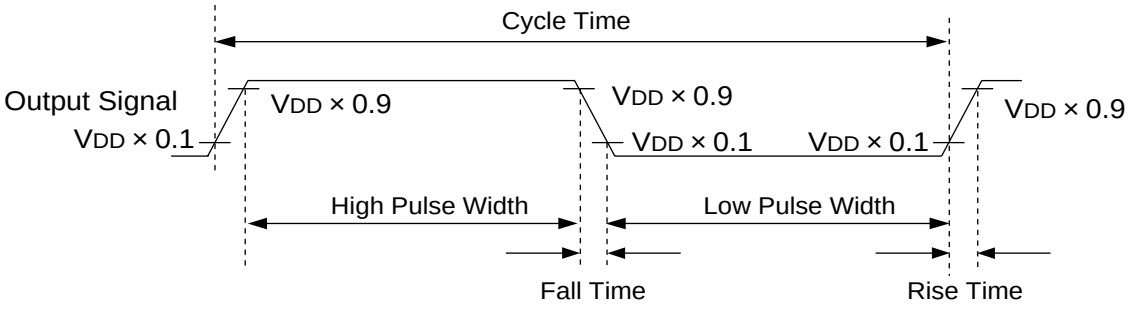
Output Signal Characteristics

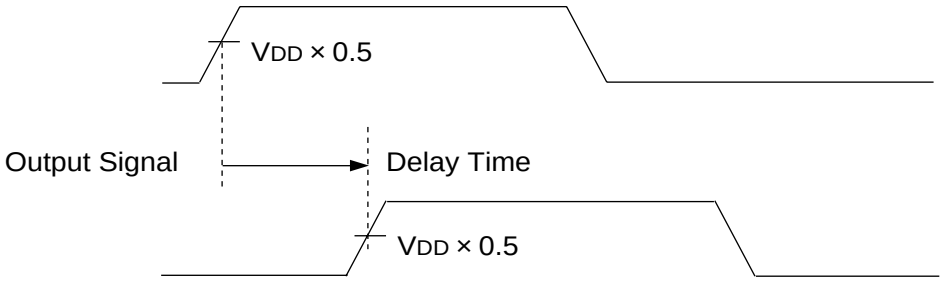
V_{DD} = 3.0 V to 3.6 V
V_{SS} = 0 V
Ta = -20 °C to +70 °C
C_L = 70 pF

Parameter		Symbol	Conditions	Capacitance			Unit
				Min	Typ	Max	
Serial interfase signal output timing (synchronous serial transmission)							
F31	Data transfer delay time (SBO3-0)	t_{TXDD}	Figure 12-1-11 Figure 12-1-12			$\frac{tcyc}{2}$	ns
F32	Data transfer hold time (during transfer) (SBO3-0)	t_{TXDH1}	Figure 12-1-11	0			ns
F33	Data transfer hold time (end of transfer at SBT input) (SBO3-0)	t_{TXDH2}	Figure 12-1-12	t_{cyc}			ns

AC Timing Voltag Level







(Both setup time and hold time are VDD × 0.5)

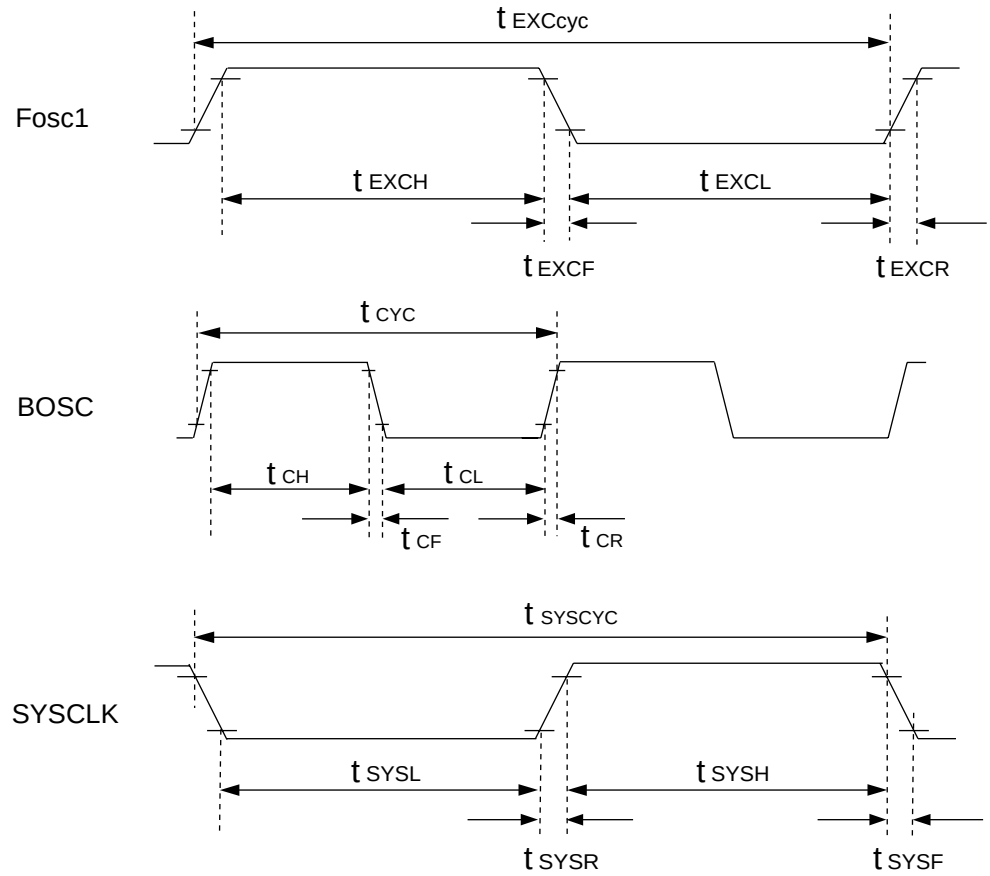


Figure 12-1-1 System Clock Timing

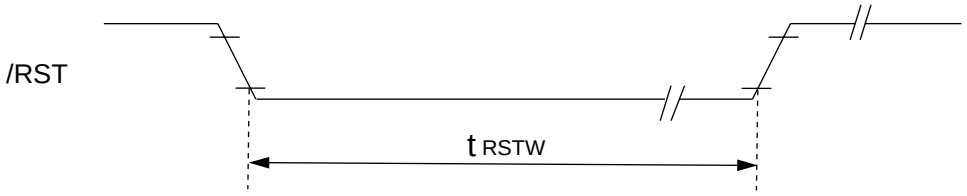


Figure 12-1-2 Reset Timing

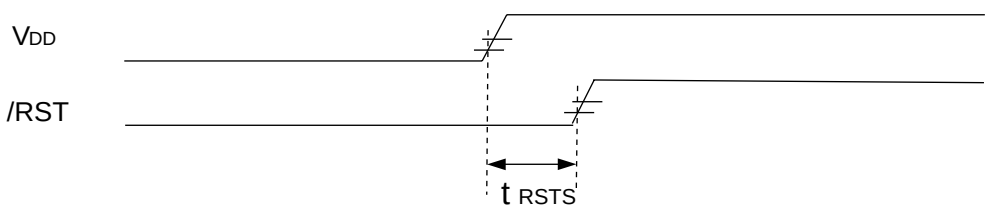


Figure 12-1-3 Power Reset Timing

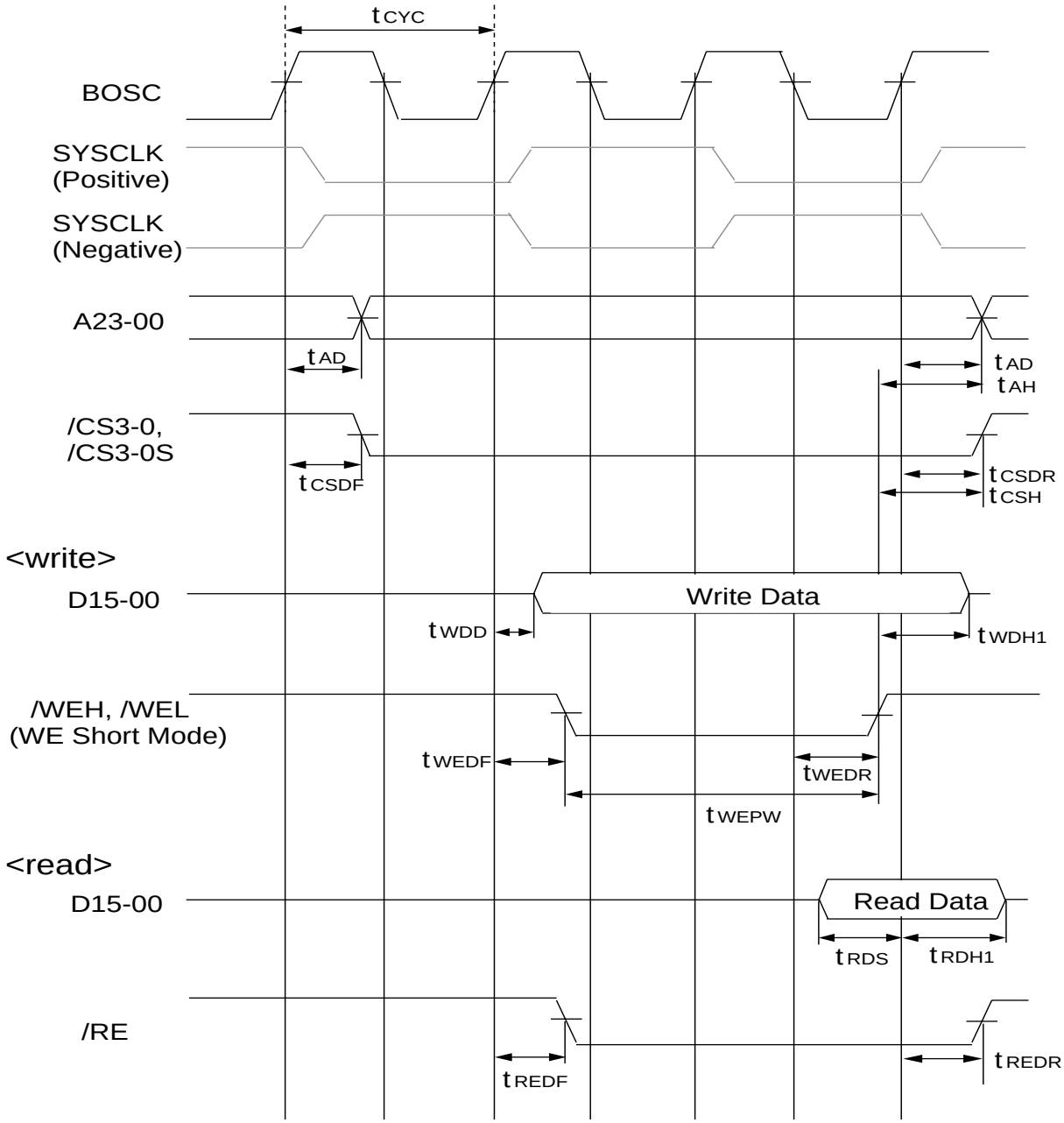


Figure 12-1-4 Data Transfer Signal Timing (0.5 Wait)

Note: Set the external wait to 0.5 or larger. (0 wait is unavailable.)

Note: Use the /WEH, /WEL signal in WE short mode.

Note: Late access mode is unavailable in the 0.5 wait access space.

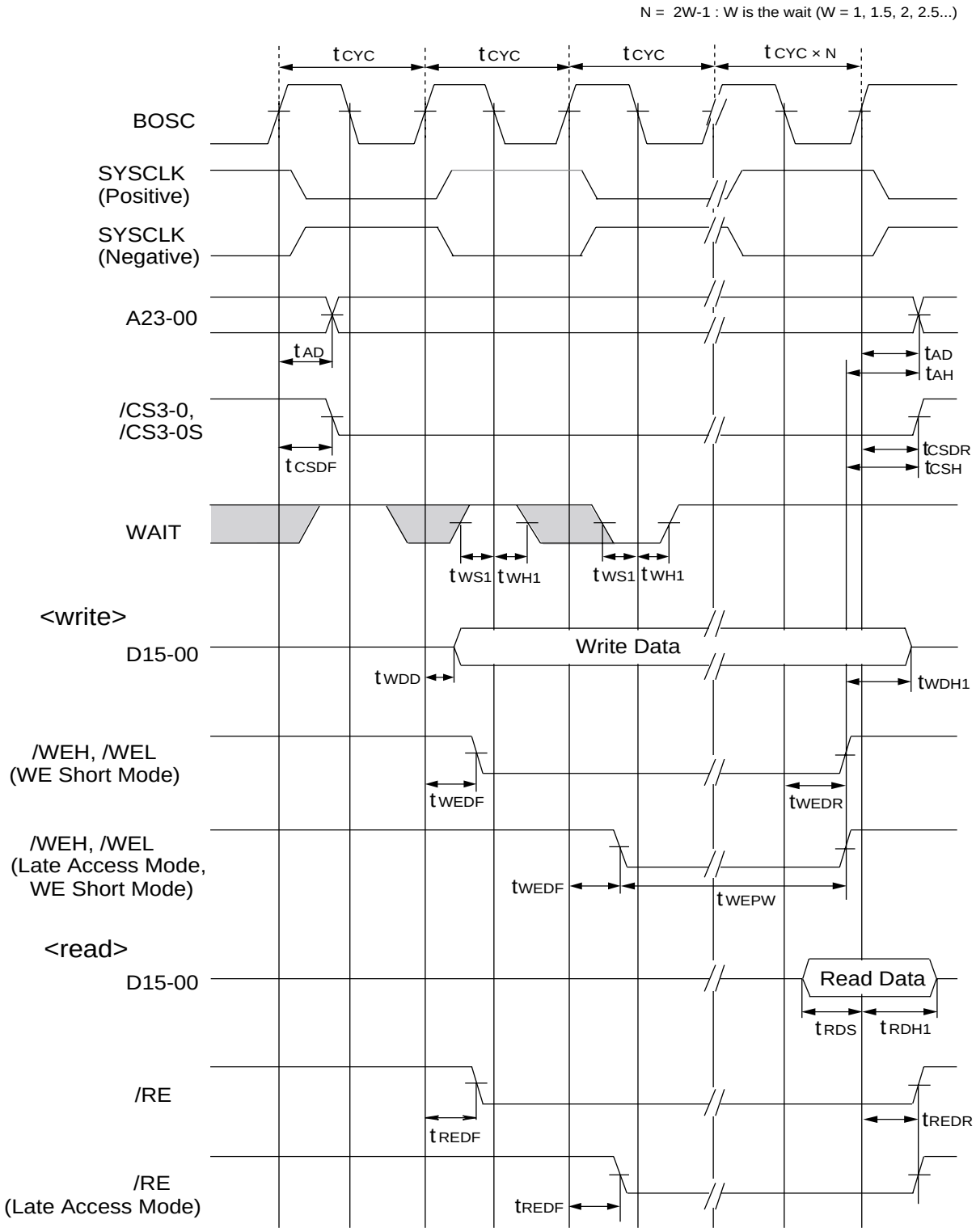


Figure 12-1-5 Data Transfer Signal Timing (1 or Larger Wait)

Note: Use the /WEH, /WEL signal in WE short mode.

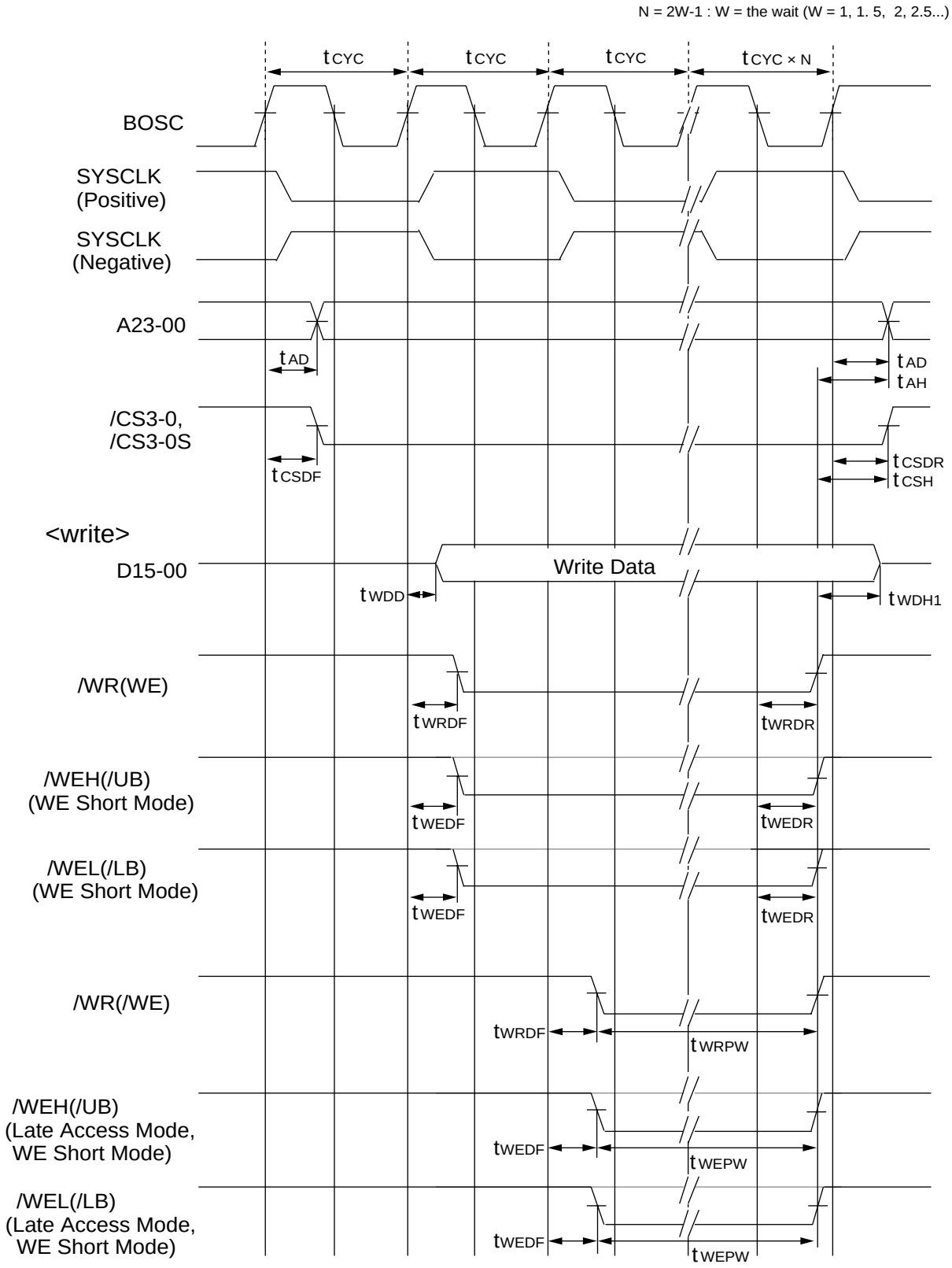


Figure 12-1-6 Data Transfer Signal Timing (Byte Data Control Mode, 1 Wait or Larger)

Note: Use the /WEH, /WEL signal in WE short mode.

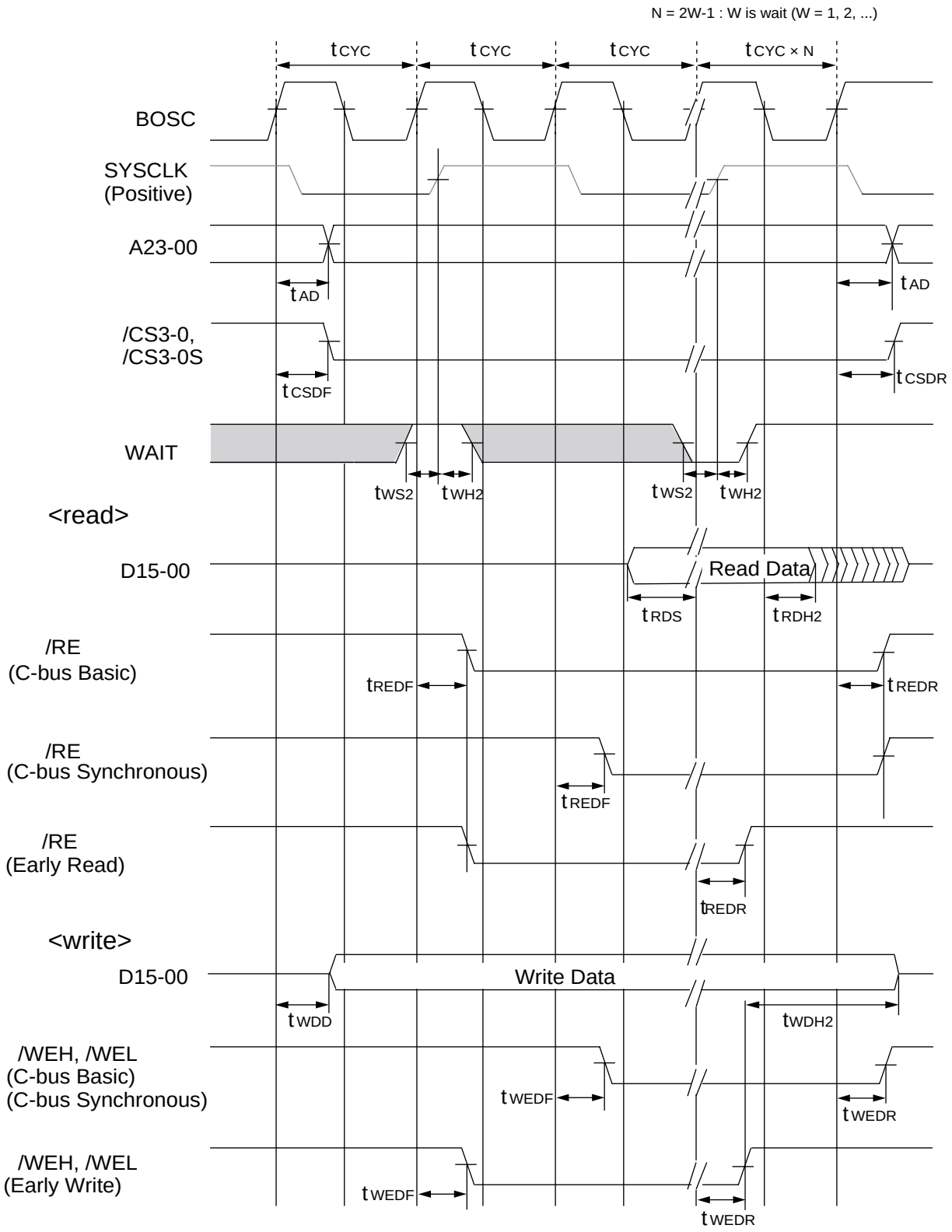


Figure 12-1-7 ATC Data Transfer Signal Timing
(C-bus Basic, C-bus Synchronous, Early Read, Early Write)

Note: Rise of /RE in early read and /WEH, /WEL in early write show the first rise of BOSC in the last tCYC cycle (on wait insert).

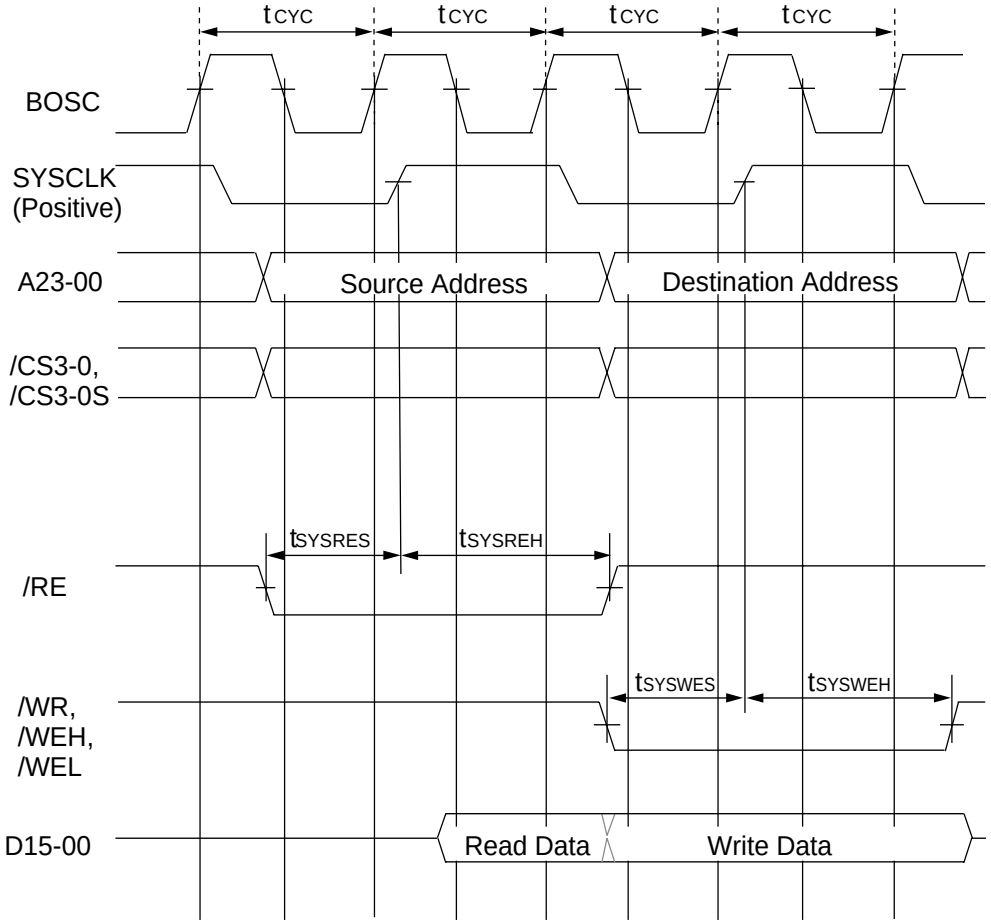


Figure 12-1-8 ATC Data Transfer Signal Timing
(1 Cycle Operation Mode)

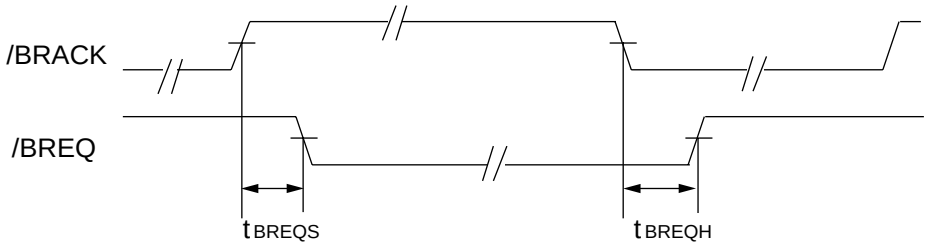


Figure 12-1-9 Bus Open Request Signal Timing

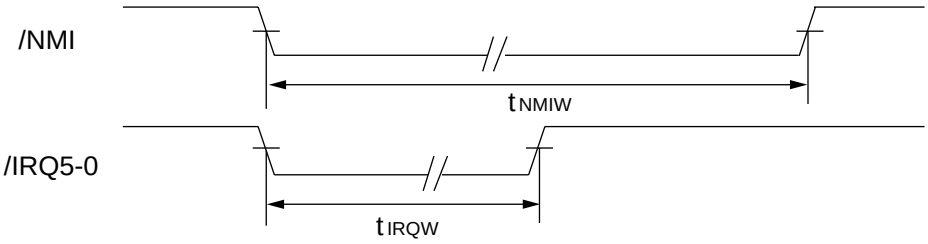


Figure 12-1-10 Interrupt Signal Input Timing

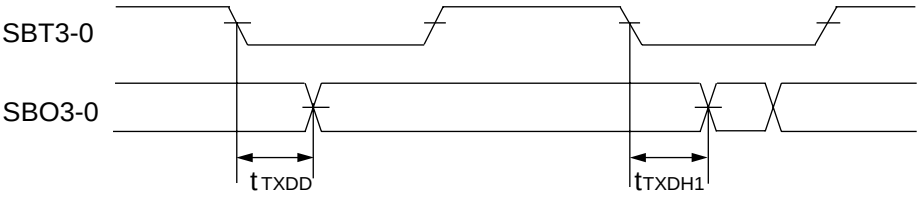


Figure 12-1-11 Serial Interface Signal Timing 1
(Synchronous Serial Transmission: During Transfer)

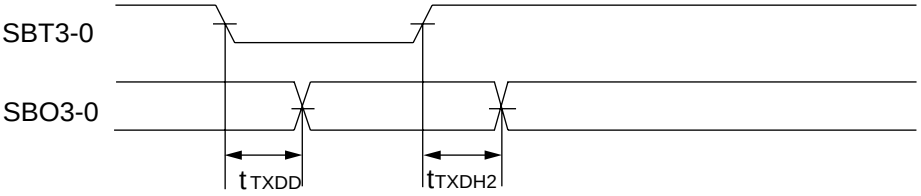


Figure 12-1-12 Serial Interface Signal Timing 2
(Synchronous Serial Transmission: End of Transfer at SBT Input)

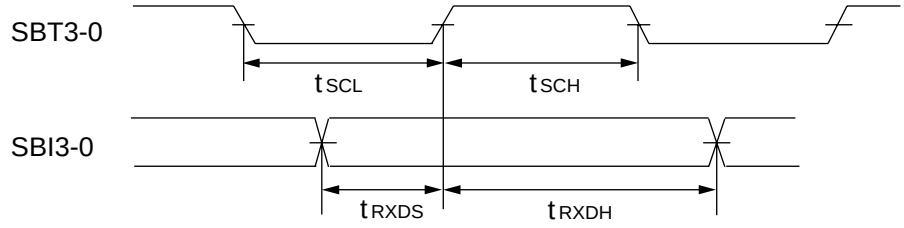


Figure 12-1-13 Serial Interface Signal Timing 3
(Synchronous Serial Reception: End of Transfer at SBT Input)

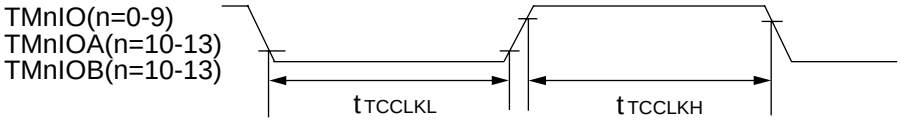


Figure 12-1-14 Timer/Counter Signal Timing

12-2 MN102H74G/74F/74D/F74G Register Address Map

Upper bits \ Lower bits	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
x'00FC00'	IAGR						EFCR ●								CPUM ●		Internal Control
x'00FC30'	MEMMD3S ●		MEMMD2S ●		MEMMD1S ●		MEMMD0S ●		MEMMD3 ●		MEMMD2 ●		MEMMD1 ●		MEMMD0 ●		Memory Control
x'00FC40'	G7ICR		G6ICR		G5ICR		G4ICR		G3ICR		G2ICR		G1ICR		G0ICR		Interruput Control
x'00FC50'					WDREG ●		EXTMD1		EXTMD0		reserved		G9ICR		G8ICR		
x'00FE00'								PS0BC							TM1BC	TM0BC	8-bit Timer
x'00FE10'								PS0BR							TM1BR	TM0BR	
x'00FE20'								PS0MD							TM1MD	TM0MD	
x'E00000'												PSCNT69				PSCNT25	8-bit Timer
x'E00010'									TM9BC	TM8BC	TM7BC	TM6BC	TM5BC	TM4BC	TM3BC	TM2BC	
x'E00020'									TM9BR	TM8BR	TM7BR	TM6BR	TM5BR	TM4BR	TM3BR	TM2BR	
x'E00030'									TM9MD	TM8MD	TM7MD	TM6MD	TM5MD	TM4MD	TM3MD	TM2MD	
x'E00100'																PSCNT10	16-bit Timer
x'E00110'							TM10BC		TM10CB		TM10CA		TM10MDB	TM10MDA	TM10MD		
x'E00120'																PSCNT11	
x'E00130'							TM11BC		TM11CB		TM11CA		TM11MDB	TM11MDA	TM11MD		
x'E00140'																PSCNT12	
x'E00150'							TM12BC		TM12CB		TM12CA		TM12MDB	TM12MDA	TM12MD		
x'E00160'																PSCNT13	
x'E00170'							TM13BC		TM13CB		TM13CA		TM13MDB	TM13MDA	TM13MD		
x'E00200'										SCA0STR1		SCA0TB	SCA0STR❍	SCA0RB❍	SCA0CTR		Serial Interface
x'E00210'										SCA1STR1		SCA1TB	SCA1STR❍	SCA1RB❍	SCA1CTR		
x'E00220'												SCA2TB	SCA2STR❍	SCA2RB❍	SCA2CTR		
x'E00230'												SCA3TB	SCA3STR❍	SCA3RB❍	SCA3CTR		
x'E00240'									reserved		reserved		reserved		reserved		
x'E00300'															ANCTR		A/D Converter
x'E00310'	AN7BUF		AN6BUF		AN5BUF		AN4BUF		AN3BUF		AN2BUF		AN1BUF		AN0BUF		
x'E00350'									reserved		reserved		reserved		reserved		
x'E00400'									AT3CTR ●		AT2CTR ●		AT1CTR ●		AT0CTR ●		ATC
x'E00410'	AT3MMDS ●		AT2MMDS ●		AT1MMDS ●		AT0MMDS ●		AT3MMD ●		AT2MMD ●		AT1MMD ●		AT0MMD ●		
x'E00420'												ATSEL		ATPRI	ATIRQ		
x'E00430'									AT3CNT ●		AT2CNT ●		AT1CNT ●		AT0CNT ●		
x'E00440'		AT3SRC1	AT3SRC0			AT2SRC1	AT2SRC0			AT1SRC1	AT1SRC0			AT0SRC1	AT0SRC0		
x'E00450'		AT3DST1	AT3DST0			AT2DST1	AT2DST0			AT1DST1	AT1DST0			AT0DST1	AT0DST0		
x'E00460'															ATRQSEL		
x'E00470'	reserved																
x'E00500'						ADBCTL		ADB1H	ADB1			ADB0H	ADB0			SYSCTL	System Control
x'E00510'		USBIE		ENDPIE		USBI		ENDPI		MAXP		INDEX		PM		FA	USB Function
x'E00520'										FN2		FN1		SBEP4		SBEP3	
x'E00530'				OWC2		OWC1		OCSR2		OCSR1		ICSR3		ICSR2		ICSR1	
x'E00540'		EP7FIFO		EP6FIFO		EP5FIFO		EP4FIFO		EP3FIFO		EP2FIFO		EP1FIFO		EP0FIFO	
x'E00550'		ATCSW														EP8FIFO	
x'E00600'	SG90ICR		SG80ICR		SG71ICR		SG70ICR		SG61ICR		SG60ICR		SG50ICR		SG40ICR		Expansion Interrupt Control
x'E00610'	SG91ICR		SG81ICR												reserved		
x'E00620'	reserved		reserved		reserved		reserved		reserved		reserved		reserved		reserved		
x'E00630'	reserved		reserved		reserved		reserved		reserved		reserved		reserved		reserved		
x'E00700'					PBPLU	PAPLU	P9PLU	P8PLU	P7PLU	P6PLU	P5PLU	P4PLU	P3PLU	P2PLU	P1PLU	P0PLU	Port Control
x'E00710'					PBOUT	PAOUT	P9OUT	P8OUT	P7OUT	P6OUT	P5OUT	P4OUT	P3OUT	P2OUT	P1OUT	P0OUT	
x'E00720'					PBIN	PAIN	P9IN	P8IN	P7IN	P6IN	P5IN	P4IN	P3IN	P2IN	P1IN	P0IN	
x'E00730'					PBDIR	PADIR	P9DIR	P8DIR	P7DIR	P6DIR	P5DIR	P4DIR	P3DIR	P2DIR	P1DIR	P0DIR	
x'E00740'			PAHMD	PALMD		P9MD	P8MD	P7MD	P6MD	P5MD	P4HMD	P4LMD	P3MD	P2MD	P1MD	P01MD	

❍ : Access by 8 bit. Use MOVB instruction.

● : Access by 16 bit in write operation. Use MOV instruction.

12-3 List of Pin Functions

EO = External Oscillation

	Pin Name	Input Level	Output Level	Schmidt	Pull-up	RESET (Note1)	RESET (Note2)	BREQ = "L"	STOP/ HALT
1	P50, WAIT	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
2	P51, /RE	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /RE	Hi-Z at /RE
3	P52, /WEL	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /WEL	Hi-Z at /WEL
4	P53, /WEH	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /WEH	Hi-Z at /WEH
5	P60, /CS0	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /CS0	Hi-Z at /CS0
6	P61, /CS1	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /CS1	Hi-Z at /CS1
7	P62, /CS2	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /CS2	Hi-Z at /CS2
8	P63, /CS3	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /CS3	Hi-Z at /CS3
9	P64, /BREQ, TM0IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Low	*
10	P65, /BRACK, TM1IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	/BRACK	*
11	P66, /WR	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at /WR	Hi-Z at /WR
12	P67, /WORD	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
13	P20, A00	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A00	Hi-Z at A00
14	P21, A01	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A01	Hi-Z at A01
15	P22, A02	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A02	Hi-Z at A02
16	P23, A03	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A03	Hi-Z at A03
17	V_{DD}	--	--	--	--	--	--	--	--
18	P54, BOSC, SYSCLK	--	CMOS	--	--	Low(BOSC)	Low(BOSC)	*	Note3
19	V_{SS}	--	--	--	--	--	--	--	--
20	XI	CMOS	--	--	--	--	--	--	--
21	XO	--	--	--	--	High(EO)	High(EO)	*	Note4
22	VDD	--	--	--	--	--	--	--	--
23	OSCI	CMOS	--	--	--	--	--	--	--
24	OSCO	--	--	--	--	High(EO)	High(EO)	*	Note5
25	MODE	CMOS	--	Yes	--	High(Input)	Low(Input)	MODE	MODE
26	P24, A04	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A04	Hi-Z at A04
27	P25, A05	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A05	Hi-Z at A05
28	P26, A06	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A06	Hi-Z at A06
29	P27, A07	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A07	Hi-Z at A07
30	P30, A08	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A08	Hi-Z at A08
31	P31, A09	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A09	Hi-Z at A09
32	P32, A10	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A10	Hi-Z at A10
33	P33, A11	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A11	Hi-Z at A11
34	V_{DD}	--	--	--	--	--	--	--	--
35	P34, A12	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A12	Hi-Z at A12
36	P35, A13	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A13	Hi-Z at A13
37	P36, A14	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A14	Hi-Z at A14
38	P37, A15	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A15	Hi-Z at A15
39	P40, A16	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A16	Hi-Z at A16
40	P41, A17	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A17	Hi-Z at A17
41	P42, A18	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A18	Hi-Z at A18
42	P43, A19, TM2IO	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A19	Hi-Z at A19
43	V_{SS}	--	--	--	--	--	--	--	--
44	P44, A20, TM3IO	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A20	Hi-Z at A20
45	P45, A21, TM4IO	TTL	CMOS	Yes	Programmable	Hi-Z	Undefined	Hi-Z at A21	Hi-Z at A21
46	P46, A22, TM5IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at A22	Hi-Z at A22
47	P47, A23, TM6IO, /CS0S	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at A2	Hi-Z at A23
48	P70, SBI3, TM7IO, /CS1S	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
49	P71, SBO3, TM8IO, /CS2S	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
50	P72, SBT3, TM9IO, /CS3S	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*

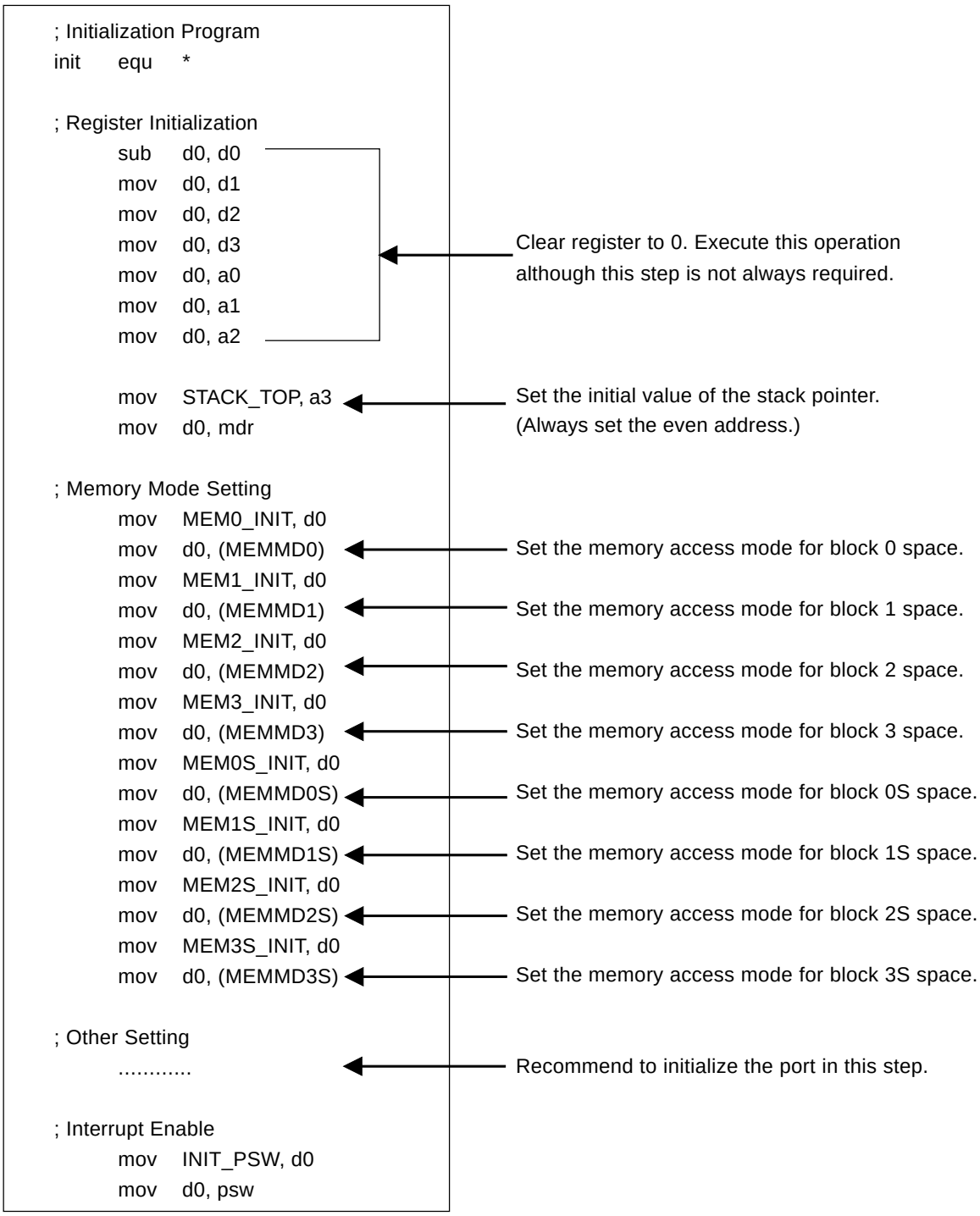
51	P80, TM10IOA, WDOUT	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
52	P81, TM10IOB, STOP	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
53	USBMODE	CMOS	--	Yes	--	USBMODE	USBMODE	USBMODE	USBMODE
54	V_{DD}	--	--	--	--	--	--	--	--
55	D+	USB Port	USB Port	--	--	D+	D+	D+	D+
56	D-	USB Port	USB Port	--	--	D-	D-	D-	D-
57	V_{SS}	--	--	--	--	--	--	--	--
58	P82, SBI2	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
59	P83, SBO2, TM11IOA	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
60	P84, SBT2, TM11IOB	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
61	V_{SS}	--	--	--	--	--	--	--	--
62	P90, AN0, TM12IOA	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
63	P91, AN1, TM12IOB	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
64	P92, AN2, TM13IOA	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
65	P93, AN3, TM13IOB	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
66	V_{DD}	--	--	--	--	--	--	--	--
67	PA0, SBI1, AN4	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
68	PA1, SBO1, AN5, SDA1	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
69	PA2, SBT1, AN6, SCL1	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
70	PA3, SBI0, AN7	Analog,TTL	CMOS	Yes(TTL)	Programmable	Hi-Z	Hi-Z	*	*
71	PA4, SBO0, SDA0	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
72	PA5, SBT0, SCL0	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
73	Pull-up	--	--	--	--	--	--	--	--
74	Pull-up	--	--	--	--	--	--	--	--
75	/NMI	CMOS	--	Yes	--	/NMI	/NMI	/NMI	/NMI
76	PB0, /IRQ0	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
77	PB1, /IRQ1	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
78	PB2, /IRQ2	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
79	PB3, /IRQ3	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
80	PB4, /IRQ4	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
81	PB5, /IRQ5	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	*	*
82	/RST	CMOS	--	Yes	--	Low(Input)	Low(Input)	High	High
83	V_{DD}	--	--	--	--	--	--	--	--
84	P00, D00	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D00	Hi-Z at D00
85	P01, D01	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D01	Hi-Z at D01
86	P02, D02	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D02	Hi-Z at D02
87	P03, D03	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D03	Hi-Z at D03
88	P04, D04	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D04	Hi-Z at D04
89	P05, D05	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D05	Hi-Z at D05
90	P06, D06	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D06	Hi-Z at D06
91	P07, D07	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D07	Hi-Z at D07
92	V_{SS}	--	--	--	--	--	--	--	--
93	P10, D08, TM2IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D08	Hi-Z at D08
94	P11, D09, TM3IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D09	Hi-Z at D09
95	P12, D10, TM4IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D10	Hi-Z at D10
96	P13, D11, TM5IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D11	Hi-Z at D11
97	P14, D12, TM6IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D12	Hi-Z at D12
98	P15, D13, TM7IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D13	Hi-Z at D13
99	P16, D14, TM8IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D14	Hi-Z at D14
100	P17, D15, TM9IO	TTL	CMOS	Yes	Programmable	Hi-Z	Hi-Z	Hi-Z at D15	Hi-Z at D15

Note1: Single-chip mode Note2: Processior mode.

Note3: Low at STOP0/1 Note4: High at STOP0/1 Note5: High at STOP0/1, HALT1

12-4 Initialization Program

The initialization program must be executed first after reset release.
The initialization program should be allocated on x'80000' in single-chip mode, memory expansion mode or processor mode.


; Initialization Program
init equ *
; Register Initialization
sub d0, d0
mov d0, d1
mov d0, d2
mov d0, d3
mov d0, a0
mov d0, a1
mov d0, a2
Clear register to 0. Execute this operation although this step is not always required.
mov STACK_TOP, a3
mov d0, mdr
Set the initial value of the stack pointer. (Always set the even address.)
; Memory Mode Setting
mov MEM0_INIT, d0
mov d0, (MEMMD0)
Set the memory access mode for block 0 space.
mov MEM1_INIT, d0
mov d0, (MEMMD1)
Set the memory access mode for block 1 space.
mov MEM2_INIT, d0
mov d0, (MEMMD2)
Set the memory access mode for block 2 space.
mov MEM3_INIT, d0
mov d0, (MEMMD3)
Set the memory access mode for block 3 space.
mov MEM0S_INIT, d0
mov d0, (MEMMD0S)
Set the memory access mode for block 0S space.
mov MEM1S_INIT, d0
mov d0, (MEMMD1S)
Set the memory access mode for block 1S space.
mov MEM2S_INIT, d0
mov d0, (MEMMD2S)
Set the memory access mode for block 2S space.
mov MEM3S_INIT, d0
mov d0, (MEMMD3S)
Set the memory access mode for block 3S space.
; Other Setting
.....
Recommend to initialize the port in this step.
; Interrupt Enable
mov INIT_PSW, d0
mov d0, psw

12-5 Flash EEPROM Version

12-5-1 Overview

The MN102HF74G replaces the MN102H74G/74F/74D mask ROM with the 128 KB Flash EEPROM which is an electrically erasable/programmable memory.

The MN102HF74G has two modes: PROM writer mode which uses a dedicated writer (our PanaX Flash Writer) and onboard serial programming mode which the CPU controls (using PanaX Flash Writer).

> PanaX Writer is a software development tool to accelerate softwares and hardwares of your set. For the mass production of your set, use a writer made of Business Partner.

The 128 KB flash memory is divided into 2 spaces as follows:

1. Load program area (2 KB: x'080000' to x'0807FF')

   This area stores the load program for serial programming. It is programmed only in PROM writer mode. It is disable to erase/write in onboard serial programming mode for the hardware condition.

> It is enable to erase/write for user program area in onboard serial programming mode of PanaX writer.

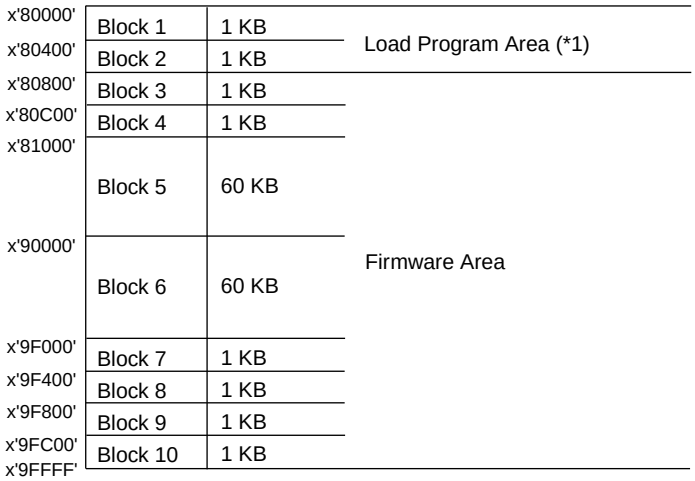
2. Firmware area (126 KB: x'080800' to x'09FFFF')

   This area stores the user program. It is programmed in both PROM writer mode and onboard serial programming mode.

The operation is guaranteed with up to 30 programming.

> A cycle of erasing to programming is counted as 1 time no matter how many blocks are rewritten. Even when the multi-block is written separately or the same block is written, each block rewrite is counted. For example, rewriting Block 1, Block 2 and Block 3 respectively is counted as 3 times. Therefore, to program efficiently, rewrite all blocks in the lump.

The following figure shows Flash EEPROM memory map.



(*1) It can erase/write as firmware area
on onboard serial programming mode of PanaX Flash Writer.

Figure 12-5-1 Flash EEPROM Memory Map

12-5-2 Flash EEPROM Programming

The following figure shows the steps of flash memory programming.

Programming starts after erasing is completed.

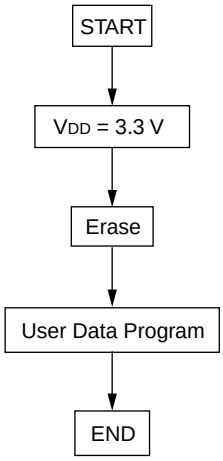


Figure 12-5-2 Flash EEPROM Program Flow

Programing must be done after erasing. Erasing process is not required for erased products released by Matsushita. Erasing process is always required before programming for products with unclear backgrounds. Even if the result of blank check of PROM writer or onboard writer is “pass”, erasing may be not enough. In this case, the reliability of the programming data is reduced even if programming ends normally. Do not program additionally for the address already programmed.

12-5-3 PROM Writer Mode

In this mode, the MN102HF74G allows a PROM writer to program the flash EEPROM.

The MN102HF74G uses a dedicated adaptor. Using the dedicated adaptor selects PROM writer mode automatically.

Check the following web page of our microcomputer division for the writer matching information.

http://www.mec.panasonic.co.jp/sc/division/micom

It isn’t compliant to the DATA-I/O’s LabSite PROM writer.

12-5-4 Onboard Serial Programming Mode

The serial programming mode is used to program the flash EEPROM in the MN102HF74G that is installed on the board.

When using our PanaX Flash Writer, the on-chip debug function enables to rewrite without the load program.

12-5-5 Connecting Onboard Serial Programming Mode

■ Connection to PanaX Flash Writer

To connect PanaX Flash Writer, The MN102HF74G need to connect 2 signal communication cable, Reset, VDD, VSS. By setting up the 10 pins flat cable connector on the target board, it makes easy to connect the Flash Writer. If it can't mount, it is possible to connect by soldering.

At reset release, all I/O pins become input pins except MODE, /RST, /IRQ5 to /IRQ0, /WORD, SBT4, SBD4, XI, OSCI.

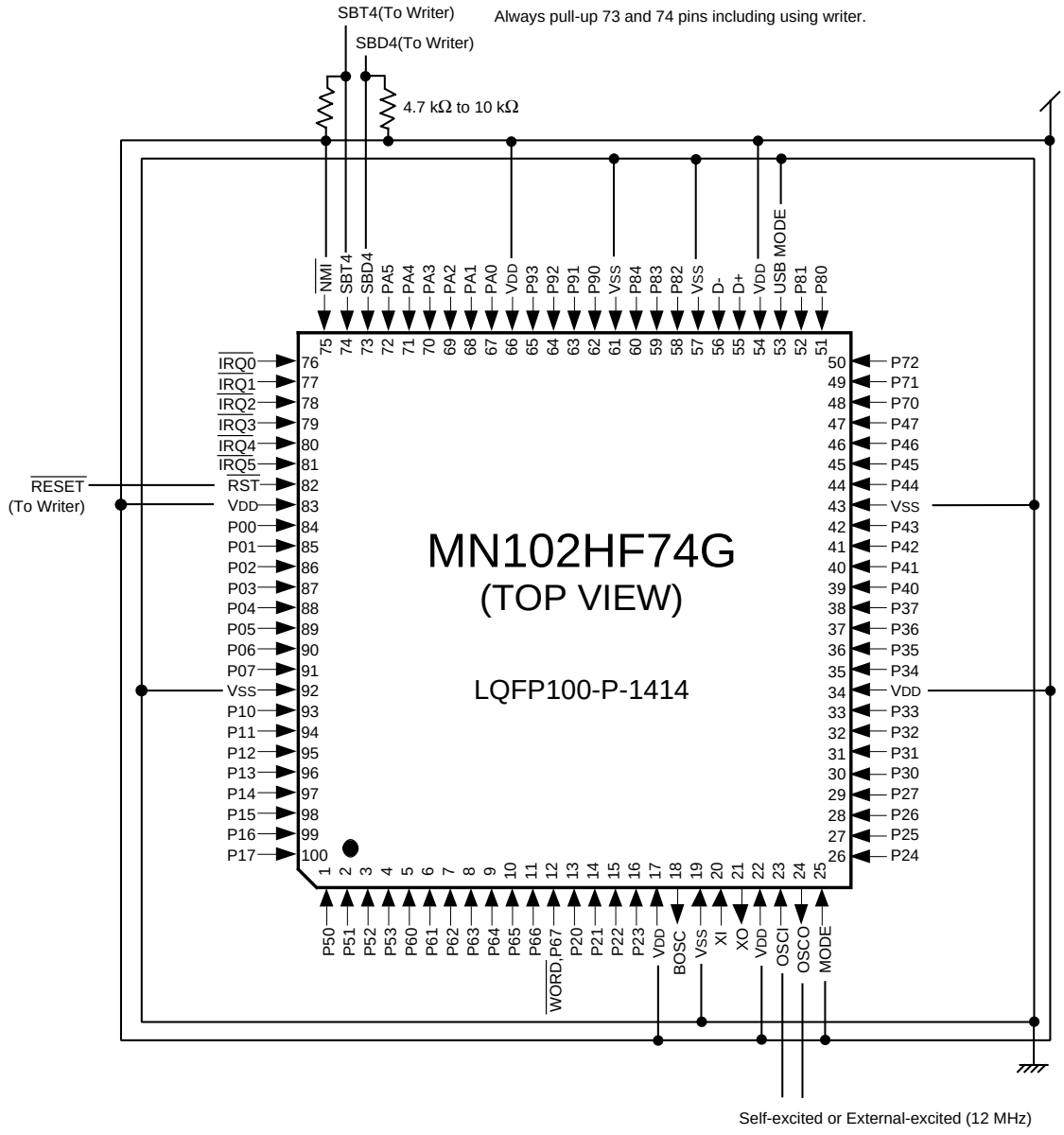


Figure 12-5-3 Pin Configuration During Serial Programming

Pins 73, 74, 82 connect to onboard serial writer. V_{DD} and V_{SS} connect to the external power sources of 3.3 V and 0 V. The level is detected by the writer, V_{DD} and V_{SS} must be output to the writer. OSCI and OSCO must be set to the self-excited oscillation or external-excited oscillation. The input pins with no specifications in the above figure are “Don’t care”. Fix them to V_{DD} or V_{SS}. The output pins with no specifications in the above figure are “OPEN”.

Table 12-5-1 Pin Connection of MN102HF74G and Flash Writer

MN102HF74G Name(No.)	Flash Writer Name(No.)	I/O	Description
/RST (82pin)	NRST (1 pin)	MN102HF74G ← → Flash Writer	Reset
SBD4 (73 pin)	TDO (3 pin)	MN102HF74G ← → Flash Writer	Communication Data
SBT4 (74 pin)	TCLK (9 pin)	MN102HF74G ← Flash Writer	Communication Clock
V_{DD}	V_{DD} (4 pin)	MN102HF74G → Flash Writer	Power of MN102HF74G
V_{SS}	GND (2, 10 pin)	-	Ground

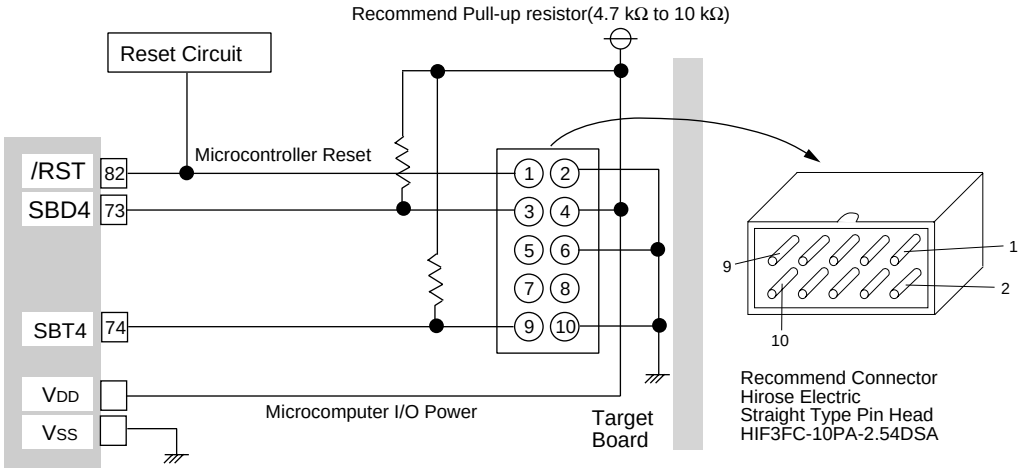


Figure 12-5-4 10 Pins Flat Cable for Interface Unit

Connecting Note

1. To onboard serial programming, the microcontroller must be in operation.
2. Pull-up process must be done for SBT4 (Pin 74), SBD4 (Pin 73) even without using Flash Writer.
3. Flash Writer outputs reset signal of the open collector. Use reset circuit of the open collector to prevent signals from confliction.

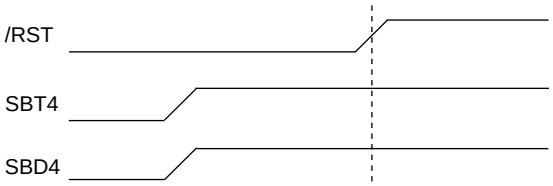
Notice the pin number of 10 pins flat cable connector. From the cable side, in notch-upper, Pin 1 is set to upper right corner and Pin 2 comes under the pin. Pin 9 is set to upper left corner and Pin 10 comes under the pin.

■ Switching SBT4 and SBD4 mode

SBT4 and SBD4 is used to programming in serial writer, however they are input pins of fixed pull-up in NORMAL mode. Transition to flash programming mode is controlled by the level of SBT4 and SBD4 at the rising edge of reset.

1) NORMAL Mode

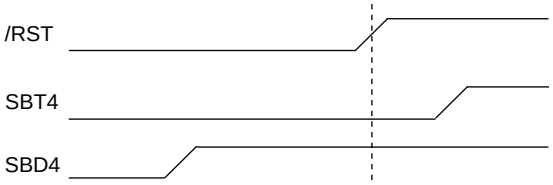
They are allocated the input pin of fixed pull-up.



When SBT4 = H, SBD4 = H at the reset rising, they come to NORMAL mode.
Do pull-up process.

2) Flash Programming Mode

They are allocated the pin for PanaX Flash Writer.



When SBT4 = L, SBD4 = H at the reset rising, they come to flash programming mode and become the dedicated pins for Flash Writer.

* Because Flash Writer controls reset, SBT4 and SBD4, microcontroller becomes flash programming mode. SBT4 and SBD4 need no particular process except pull-up.

MN102H SERIES INSTRUCTION SET

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
MOV	MOV Dm,An	Dm→An	—	—	—	—	—	—	—	—	—	2	2	F2:30+Dm<<2+An
	MOV An,Dm	An→Dm	—	—	—	—	—	—	—	—	—	2	2	F2:F0+An<<2+Dm
	MOV Dn,Dm	Dn→Dm	—	—	—	—	—	—	—	—	—	1	1	80+Dn<<2+Dm *1
	MOV An,Am	An→Am	—	—	—	—	—	—	—	—	—	2	2	F2:70+An<<2+Am
	MOV PSW,Dn	PSW→Dn	0	—	—	—	—	—	—	—	—	2	2	F3:F0+Dn
	MOV Dn,PSW	Dn→PSW	—	●	●	●	●	●	●	●	●	2	3	F3:D0+Dn<<2 *2
	MOV MDR,Dn	MDR→Dn	0	—	—	—	—	—	—	—	—	2	2	F3:E0+Dn
	MOV Dn,MDR	Dn→MDR	—	—	—	—	—	—	—	—	—	2	2	F3:C0+Dn<<2
	MOV (An),Dm	mem16(An)→Dm	S	—	—	—	—	—	—	—	—	1	1	20+An<<2+Dm
	MOV (d8,An),Dm	mem16(An+d8)→Dm	S	—	—	—	—	—	—	—	—	2	1	60+An<<2+Dm:d8
	MOV (d16,An),Dm	mem16(An+d16)→Dm	S	—	—	—	—	—	—	—	—	4	2	F7:C0+An<<2+Dm:d16-l:d16-h
	MOV (d24,An),Dm	mem16(An+d24)→Dm	S	—	—	—	—	—	—	—	—	5	3	F4:80+An<<2+Dm:d24-l:d24-m:d24-h
	MOV (Di,An),Dm	mem16(An+Di)→Dm	S	—	—	—	—	—	—	—	—	2	2	F1:40+Di<<4+An<<2+Dm
	MOV (abs16),Dn	mem16(abs16)→Dn	S	—	—	—	—	—	—	—	—	3	1	C8+Dn:abs16-l:abs16-h
	MOV (abs24),Dn	mem16(abs24)→Dn	S	—	—	—	—	—	—	—	—	5	3	F4:C0+Dn:abs24-l:abs24-m:abs24-h
	MOV (An),Am	mem24(An)→Am	—	—	—	—	—	—	—	—	—	2	2	70+An<<2+Am:00 *3
	MOV (d8,An),Am	mem24(An+d8)→Am	—	—	—	—	—	—	—	—	—	2	2	70+An<<2+Am:d8
	MOV (d16,An),Am	mem24(An+d16)→Am	—	—	—	—	—	—	—	—	—	4	3	F7:B0+An<<2+Am:d16-l:d16-h
	MOV (d24,An),Am	mem24(An+d24)→Am	—	—	—	—	—	—	—	—	—	5	4	F4:F0+An<<2+Am:d24-l:d24-m:d24-h
	MOV (abs16),An	mem24(abs16)→An	—	—	—	—	—	—	—	—	—	4	3	F7:30+An:abs16-l:abs16-h
	MOV (abs24),An	mem24(abs24)→An	—	—	—	—	—	—	—	—	—	5	4	F4:D0+An:abs24-l:abs24-m:abs24-h
	MOV Dm,(An)	Dm→mem16(An)	—	—	—	—	—	—	—	—	—	1	1	00+An<<2+Dm
	MOV Dm,(d8,An)	Dm→mem16(An+d8)	—	—	—	—	—	—	—	—	—	2	1	40+An<<2+Dm:d8
	MOV Dm,(d16,An)	Dm→mem16(An+d16)	—	—	—	—	—	—	—	—	—	4	2	F7:80+An<<2+Dm:d16-l:d16-h
	MOV Dm,(d24,An)	Dm→mem16(An+d24)	—	—	—	—	—	—	—	—	—	5	3	F4:00+An<<2+Dm:d24-l:d24-m:d24-h
	MOV Dm,(Di,An)	Dm→mem16(An+Di)	—	—	—	—	—	—	—	—	—	2	2	F1:C0+Di<<4+An<<2+Dm
	MOV Dn,(abs16)	Dn→mem16(abs16)	—	—	—	—	—	—	—	—	—	3	1	C0+Dn:abs16-l:abs16-h
	MOV Dn,(abs24)	Dn→mem16(abs24)	—	—	—	—	—	—	—	—	—	5	3	F4:40+Dn:abs24-l:abs24-m:abs24-h
	MOV Am,(An)	Am→mem24(An)	—	—	—	—	—	—	—	—	—	2	2	50+An<<2+Am:00 *4
	MOV Am,(d8,An)	Am→mem24(An+d8)	—	—	—	—	—	—	—	—	—	2	2	50+An<<2+Am:d8
	MOV Am,(d16,An)	Am→mem24(An+d16)	—	—	—	—	—	—	—	—	—	4	3	F7:A0+An<<2+Am:d16-l:d16-h
	MOV Am,(d24,An)	Am→mem24(An+d24)	—	—	—	—	—	—	—	—	—	5	4	F4:10+An<<2+Am:d24-l:d24-m:d24-h
	MOV An,(abs16)	An→mem24(abs16)	—	—	—	—	—	—	—	—	—	4	3	F7:20+An:abs16-l:abs16-h
	MOV An,(abs24)	An→mem24(abs24)	—	—	—	—	—	—	—	—	—	5	4	F4:50+An:abs24-l:abs24-m:abs24-h
	MOV imm8,Dn	imm8→Dn	S	—	—	—	—	—	—	—	—	2	1	80+Dn<<2+Dn:imm8
	MOV imm16,Dn	imm16→Dn	S	—	—	—	—	—	—	—	—	3	1	F8+Dn:imm16-l:imm16-h
	MOV imm24,Dn	imm24→Dn	—	—	—	—	—	—	—	—	—	5	3	F4:70+Dn:imm24-l:imm24-m:imm24-h
	MOV imm16,An	imm16→An	0	—	—	—	—	—	—	—	—	3	1	DC+An:imm16-l:imm16-h
	MOV imm24,An	imm24→An	—	—	—	—	—	—	—	—	—	5	3	F4:74+An:imm24-l:imm24-m:imm24-h
MOVX	MOVX (d8,An),Dm	mem24(An+d8)→Dm	—	—	—	—	—	—	—	—	—	3	3	F5:70+An<<2+Dm:d8
	MOVX (d16,An),Dm	mem24(An+d16)→Dm	—	—	—	—	—	—	—	—	—	4	3	F7:70+An<<2+Dm:d16-l:d16-h
	MOVX (d24,An),Dm	mem24(An+d24)→Dm	—	—	—	—	—	—	—	—	—	5	4	F4:B0+An<<2+Dm:d24-l:d24-m:d24-h
	MOVX Dm,(d8,An)	Dm→mem24(An+d8)	—	—	—	—	—	—	—	—	—	3	3	F5:50+An<<2+Dm:d8
	MOVX Dm,(d16,An)	Dm→mem24(An+d16)	—	—	—	—	—	—	—	—	—	4	3	F7:60+An<<2+Dm:d16-l:d16-h
	MOVX Dm,(d24,An)	Dm→mem24(An+d24)	—	—	—	—	—	—	—	—	—	5	4	F4:30+An<<2+Dm:d24-l:d24-m:d24-h
MOVB	MOVB (An),Dm	mem8(An)→Dm	S	—	—	—	—	—	—	—	—	2	2	30+An<<2+Dm:B8+Dn *5
	MOVB (d8,An),Dm	mem8(An+d8)→Dm	S	—	—	—	—	—	—	—	—	3	2	F5:20+An<<2+Dm:d8
	MOVB (d16,An),Dm	mem8(An+d16)→Dm	S	—	—	—	—	—	—	—	—	4	2	F7:D0+An<<2+Dm:d16-l:d16-h
	MOVB (d24,An),Dm	mem8(An+d24)→Dm	S	—	—	—	—	—	—	—	—	5	3	F4:A0+An<<2+Dm:d24-l:d24-m:d24-h
	MOVB (Di,An),Dm	mem8(An+Di)→Dm	S	—	—	—	—	—	—	—	—	2	2	F0:40+Di<<4+An<<2+Dm
	MOVB (abs16),Dn	mem8(abs16)→Dn	S	—	—	—	—	—	—	—	—	4	2	CC+Dn:abs16-l:abs16-h:B8+Dn *6
	MOVB (abs24),Dn	mem8(abs24)→Dn	S	—	—	—	—	—	—	—	—	5	3	F4:C4+Dn:abs24-l:abs24-m:abs24-h
	MOVB Dm,(An)	Dm→mem8(An)	—	—	—	—	—	—	—	—	—	1	1	10+Dm<<2+An
	MOVB Dm,(d8,An)	Dm→mem8(An+d8)	—	—	—	—	—	—	—	—	—	3	2	F5:10+An<<2+Dm:d8
	MOVB Dm,(d16,An)	Dm→mem8(An+d16)	—	—	—	—	—	—	—	—	—	4	2	F7:90+An<<2+Dm:d16-l:d16-h
	MOVB Dm,(d24,An)	Dm→mem8(An+d24)	—	—	—	—	—	—	—	—	—	5	3	F4:20+An<<2+Dm:d24-l:d24-m:d24-h
	MOVB Dm,(Di,An)	Dm→mem8(An+Di)	—	—	—	—	—	—	—	—	—	2	2	F0:C0+Di<<4+An<<2+Dm

Notes: *1 Dn cannot be the same as Dm.
*2 In case of using this instruction, a value of PSW or the one that has been pushed in stack should not referred in the interrupt processing program.
*3 This instruction is supported by the assembler. The assembler will generate the bit pattern for "MOV(d8,An),Am"(d8=0).
*4 This instruction is supported by the assembler. The assembler will generate the bit pattern for "MOV Am,(d8,An)"(d8=0).
*5 This instruction is supported by the assembler. The assembler will generate the bit pattern for the two instructions "MOVBU(An),Dm and "EXTXB Dm".
*6 This instruction is supported by the assembler. The assembler will generate the bit pattern for the two instructions "MOVBU(abs16),Dm and "EXTXB Dn".

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
MOVB	MOVB Dn,(abs16)	Dn→mem8(abs16)	—	—	—	—	—	—	—	—	—	3	1	C4+Dn:abs16-l:abs16-h
	MOVB Dn,(abs24)	Dn→mem8(abs24)	—	—	—	—	—	—	—	—	—	5	3	F4:44+Dn:abs24-l:abs24-m:abs24-h
MOVBU	MOVBU (An),Dm	mem8(An)→Dm	0	—	—	—	—	—	—	—	—	1	1	30+An<<2+Dm
	MOVBU (d8,An),Dm	mem8(An+d8)→›Dm	0	—	—	—	—	—	—	—	—	3	2	F5:30+An<<2+Dm:d8
	MOVBU (d16,An),Dm	mem8(An+d16)→Dm	0	—	—	—	—	—	—	—	—	4	2	F7:50+An<<2+Dm:d16-l:d16-h
	MOVBU (d24,An),Dm	mem8(An+d24)→Dm	0	—	—	—	—	—	—	—	—	5	3	F4:90+An<<2+Dm:d24-l:d24-m:d24-h
	MOVBU (Di,An),Dm	mem8(An+Di)→Dm	0	—	—	—	—	—	—	—	—	2	2	F0:80+Di<<4+An<<2+Dm
	MOVBU (abs16),Dn	mem8(abs16)→Dn	0	—	—	—	—	—	—	—	—	3	1	CC+Dn:abs16-l:abs16-h
	MOVBU (abs24),Dn	mem8(abs24)→Dn	0	—	—	—	—	—	—	—	—	5	3	F4:C8+Dn:abs24-l:abs24-m:abs24-h
EXT	EXT Dn	If Dn.bp15=0, x'0000'→MDR If Dn.bp15=1, x'FFFF'→MDR	S	—	—	—	—	—	—	—	—	2	3	F3:C1+Dn<<2 *7
EXTX	EXTX Dn	If Dn.bp15=0, Dn&x'00FFFF'→Dn If Dn.bp15=1, Dn l x'FF0000'→Dn	S	—	—	—	—	—	—	—	—	1	1	B0+Dn *8
EXTXU	EXTXU Dn	Dn&x'00FFFF→Dn	0	—	—	—	—	—	—	—	—	1	1	B4+Dn *9
EXTXB	EXTXB Dn	If Dn.bp7=0, Dn&x'0000FF'→Dn If Dn.bp7=1, Dn l x'FFFF00'→Dn	S	—	—	—	—	—	—	—	—	1	1	B8+Dn *10
EXTXBU	EXTXBU Dn	Dn&x'0000FF'→Dn	0	—	—	—	—	—	—	—	—	1	1	BC+Dn *11
ADD	ADD Dn,Dm	Dm+Dn→Dm	—	●	●	●	●	●	●	●	●	1	1	90+Dn<<2+Dm
	ADD Dm,An	An+Dm→An	—	●	●	●	●	●	●	●	●	2	2	F2:00+Dm<<2+An
	ADD An,Dm	Dm+An→Dm	—	●	●	●	●	●	●	●	●	2	2	F2:C0+An<<2+Dm
	ADD An,Am	Am+An→Am	—	●	●	●	●	●	●	●	●	2	2	F2:40+An<<2+Am
	ADD imm8,Dn	Dn+imm8→Dn	S	●	●	●	●	●	●	●	●	2	1	D4+Dn:imm8
	ADD imm16,Dn	Dn+imm16→Dn	S	●	●	●	●	●	●	●	●	4	2	F7:18+Dn:imm16-l:imm16-h
	ADD imm24,Dn	Dn+imm24→Dn	—	●	●	●	●	●	●	●	●	5	3	F4:60+Dn:imm24-l:imm24-m:imm24-h
	ADD imm8,An	An+imm8→An	S	●	●	●	●	●	●	●	●	2	1	D0+An:imm8
	ADD imm16,An	An+imm16→An	S	●	●	●	●	●	●	●	●	4	2	F7:08+An:imm16-l:imm16-h
	ADD imm24,An	An+imm24→An	—	●	●	●	●	●	●	●	●	5	3	F4:64+An:imm24-l:imm24-m:imm24-h
ADDC	ADDC Dn,Dm	Dm+Dn+CF→Dm	—	●	●	●	●	●	●	●	●	2	2	F2:80+Dn<<2+Dm
ADDNF	ADDNF imm8,An	An+imm8→An	S	—	—	—	—	—	—	—	—	3	2	F5:0C+An:imm8 *12
SUB	SUB Dn,Dm	Dm-Dn→Dm	—	●	●	●	●	●	●	●	●	1	1	A0+Dn<<2+Dm
	SUB Dm,An	An-Dm→An	—	●	●	●	●	●	●	●	●	2	2	F2:10+Dm<<2+An
	SUB An,Dm	Dm-An→Dm	—	●	●	●	●	●	●	●	●	2	2	F2:D0+An<<2+Dm
	SUB An,Am	Am-An→Am	—	●	●	●	●	●	●	●	●	2	2	F2:50+An<<2+Am
	SUB imm16,Dn	Dn-imm16→Dn	S	●	●	●	●	●	●	●	●	4	2	F7:1C+Dn:imm16-l:imm16-h
	SUB imm24,Dn	Dn-imm24→Dn	—	●	●	●	●	●	●	●	●	5	3	F4:68+Dn:imm24-l:imm24-m:imm24-h
	SUB imm16,An	An-imm16→An	S	●	●	●	●	●	●	●	●	4	2	F7:0C+An:imm16-l:imm16-h
	SUB imm24,An	An-imm24→An	—	●	●	●	●	●	●	●	●	5	3	F4:6C+An:imm24-l:imm24-m:imm24-h
SUBC	SUBC Dn,Dm	Dm-Dn-CF→Dm	—	●	●	●	●	●	●	●	●	2	2	F2:90+Dn<<2+Dm
MUL	MUL Dn,Dm	Dm ✻ Dn→Dm (Dm ✻ Dn)>>16→MDR	—	?	?	?	?	0	?	●	●	2	12	F3:40+Dn<<2+Dm *13
MULU	MULU Dn,Dm	Dm ✻ Dn→Dm (Dm ✻ Dn)>>16→MDR	—	?	?	?	?	0	?	●	●	2	12	F3:50+Dn<<2+Dm *14
MULQ	MULQ Dn,Dm	Dm ✻ Dn→Dm (H.M.) (Dm ✻ Dn)>>16→MDR	—	—	—	—	—	—	—	—	—	3	3	F5:60+Dn<<2+Dm:10
MULQL	MULQL Dn,Dm	Dm ✻ Dn→Dm (H.M.)	—	—	—	—	—	—	—	—	—	3	2	F5:40+Dn<<2+Dm:00
	MULQL imm8,Dn	Dn ✻ imm8→Dn (H.M.)	—	—	—	—	—	—	—	—	—	4	2	F5:F0+Dn:04:imm8
	MULQL imm16,Dn	Dn ✻ imm16→Dn (H.M.)	—	—	—	—	—	—	—	—	—	5	3	F5:F4+Dn:08:imm16-l:imm16-h
MULQH	MULQH Dn,Dm	(Dm ✻ Dn)>>16→Dm (H.M.)	S	—	—	—	—	—	—	—	—	3	2	F5:40+Dn<<2+Dm:01
	MULQH imm8,Dn	(Dn ✻ imm8)>>16→Dn (H.M.)	S	—	—	—	—	—	—	—	—	4	2	F5:F0+Dn:05:imm8
	MULQH imm16,Dn	(Dn ✻ imm16)>>16→Dn (H.M.)	S	—	—	—	—	—	—	—	—	5	3	F5:F4+Dn:09:imm16-l:imm16-h
DIVU	DIVU Dn,Dm	(MDR<<16+Dm)/Dn→Dm ...MDR	—	?	?	0/?	●/?	0/1	?	●/?	●/?	2	13	F3:60+Dn<<2+Dm *15

Notes: 7* 32-bit sign extended word data
8* 24-bit sign extended word data
9* 24-bit zero extended word data
10* 24-bit sign extended byte data
11* 24-bit zero extended byte data
12* Addition without changing flag
13* 16x16 = 32 (signed)
14* 16x16 = 32 (unsigned)
15* 32÷16 = 16...16 (unsigned)

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
CMP	CMP Dn,Dm	Dm-Dn...PSW	—	●	●	●	●	●	●	●	●	2	2	F3:90+Dn<<2+Dm
	CMP Dm,An	An-Dm...PSW	—	●	●	●	●	●	●	●	●	2	2	F2:20+Dm<<2+An
	CMP An,Dm	Dm-An...PSW	—	●	●	●	●	●	●	●	●	2	2	F2:E0+An<<2+Dm
	CMP An,Am	Am-An...PSW	—	●	●	●	●	●	●	●	●	2	2	F2:60+An<<2+Am
	CMP imm8,Dn	Dn-imm8...PSW	S	●	●	●	●	●	●	●	●	2	1	D8+Dn:imm8
	CMP imm16,Dn	Dn-imm16...PSW	S	●	●	●	●	●	●	●	●	4	2	F7:48+Dn:imm16-l:imm16-h
	CMP imm24,Dn	Dn-imm24...PSW	—	●	●	●	●	●	●	●	●	5	3	F4:78+Dn:imm24-l:imm24-m:imm24-h
	CMP imm16,An	An-imm16...PSW	0	●	●	●	●	●	●	●	●	3	1	EC+An:imm16-l:imm16-h
	CMP imm24,An	An-imm24...PSW	—	●	●	●	●	●	●	●	●	5	3	F4:7C+An:imm24-l:imm24-m:imm24-h
AND	AND Dn,Dm	Dm&(x'FF0000' I Dn)→Dm	—	—	—	—	—	0	0	●	●	2	2	F3:00+Dn<<2+Dm *16
	AND imm8,Dn	Dn&(x'FF0000' I imm8)→Dn	0	—	—	—	—	0	0	●	●	3	2	F5:00+Dn:imm8 *16
	AND imm16,Dn	Dn&(x'FF0000' I imm16)→Dn	—	—	—	—	—	0	0	●	●	4	2	F7:00+Dn:imm16-l:imm16-h *16
	AND imm16,PSW	PSW&imm16→PSW	—	●	●	●	●	●	●	●	●	4	3	F7:10:imm16-l:imm16-h *16
OR	OR Dn,Dm	Dm I (Dn&x'00FFFF')→Dm	—	—	—	—	—	0	0	●	●	2	2	F3:10+Dn<<2+Dm *16
	OR imm8,Dn	Dn I imm8→Dn	0	—	—	—	—	0	0	●	●	3	2	F5:08+Dn:imm8 *16
	OR imm16,Dn	Dn I imm16→Dn	—	—	—	—	—	0	0	●	●	4	2	F7:40+Dn:imm16-l:imm16-h *16
	OR imm16,PSW	PSW I imm16→PSW	—	●	●	●	●	●	●	●	●	4	3	F7:14:imm16-l:imm16-h *16
XOR	XOR Dn,Dm	Dm^(x'00FFFF'&Dn)→Dm	—	—	—	—	—	0	0	●	●	2	2	F3:20+Dn<<2+Dm *16
	XOR imm16,Dn	Dn^imm16→Dn	—	—	—	—	—	0	0	●	●	4	2	F7:4C+Dn:imm16-l:imm16-h *16
NOT	NOT Dn	Dn^x'00FFFF'→Dn	—	—	—	—	—	0	0	●	●	2	2	F3:E4+Dn *16
ASR	ASR Dn	Dn.lsb→CF Dn.bp→Dn.bp-1(bp15~1) Dn.bp15→Dn.bp15	—	—	—	—	—	0	●	●	●	2	2	F3:38+Dn *16
LSR	LSR Dn	Dn.lsb→CF Dn.bp→Dn.bp-1(bp15~1) 0→Dn.bp15	—	—	—	—	—	0	●	0	●	2	2	F3:3C+Dn *16
ROR	ROR Dn	Dn.lsb→temp Dn.bp→Dn.bp-1(bp15~1) CF→Dn.bp15 temp→CF	—	—	—	—	—	0	●	●	●	2	2	F3:34+Dn *16
ROL	ROL Dn	Dn.bp15→temp Dn.bp→Dn.bp+1(bp14~0) CF→Dn.lsb temp→CF	—	—	—	—	—	0	●	●	●	2	2	F3:30+Dn *16
BTST	BTST imm8,Dn	Dn&imm8...PSW	0	—	—	—	—	0	0	0	●	3	2	F5:04+Dn:imm8
	BTST imm16,Dn	Dn&imm16...PSW	0	—	—	—	—	0	0	●	●	4	2	F7:04+Dn:imm16-l:imm16-h
BSET	BSET Dm,(An)	mem8(An)&Dm...PSW mem8(An) I Dm→mem8(An)	0	—	—	—	—	0	0	0	●	2	5	F0:20+An<<2+Dm *17
	BSET imm8,(abs16)	mem8(abs16) I imm8 →mem8(abs16)	—	—	—	—	—	—	—	—	—	5	4	F4:E3:abs16-l:abs16-h:imm8
	BSET imm8,(abs24)	mem8(abs24) I imm8 →mem8(abs24)	—	—	—	—	—	—	—	—	—	6	5	F4:4B:abs24-l:abs24-m:abs24-h:imm8
	BSET imm8,(d8,An)	mem8(An+d8) I imm8 →mem8(An+d8)	—	—	—	—	—	—	—	—	—	4	4	F4:E8+An:d8:imm8
	BSET (abs16)bp	mem8(abs16) I (1<<bp) →mem8(abs16)	—	—	—	—	—	—	—	—	—	4	4	F5:D0+bp:abs16-l:abs16-h
	BSET (abs24)bp	mem8(abs24) I (1<<bp) →mem8(abs24)	—	—	—	—	—	—	—	—	—	6	6	F3:FE:D0+bp:abs24-l:abs24-m:abs24-h
	BSET (d8,An)bp	mem8(An+d8) I (1<<bp) →mem8(An+d8)	—	—	—	—	—	—	—	—	—	3 3 4 4	4 4 5 5	F5:90+bp:d8 An=A0 F5:98+bp:d8 An=A1 F3:FF:90+bp:d8 An=A2 F3:FF:98+bp:d8 An=A3

Notes: 16* 16-bit computation
17* Performed under the conditions of bus lock and disabled interrupts.

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
BCLR	BCLR Dm,(An)	mem8(An)&Dm...PSW mem8(An)&(~Dm)→mem8(An)	0	—	—	—	—	0	0	0	●	2	5	F0:30+An<<2+Dm *17
	BCLR imm8,(abs16)	mem8(abs16)&(~imm8) →mem8(abs16)	—	—	—	—	—	—	—	—	—	5	4	F4:E7:abs16-l:abs16-h:imm8
	BCLR imm8,(abs24)	mem8(abs24)&(~imm8) →mem8(abs24)	—	—	—	—	—	—	—	—	—	6	5	F4:4F:abs24-l:abs24-m:abs24-h:imm8
	BCLR imm8,(d8,An)	mem8(An+d8)&(~imm8) →mem8(An+d8)	—	—	—	—	—	—	—	—	—	4	4	F4:EC+An:d8:imm8
	BCLR (abs16)bp	mem8(abs16)&~(1<<bp) →mem8(abs16)	—	—	—	—	—	—	—	—	—	4	4	F5:D8+bp:abs16-l:abs16-h
	BCLR (abs24)bp	mem8(abs24)&~(1<<bp) →mem8(abs24)	—	—	—	—	—	—	—	—	—	6	6	F3:FE:D8+bp:abs24-l:abs24-m:abs24-h
	BCLR (d8,An)bp	mem8(An+d8)&~(1<<bp) →mem8(An+d8)	—	—	—	—	—	—	—	—	—	3 3 4 4	4 4 5 5	F5:B0+bp:d8 An=A0 F5:B8+bp:d8 An=A1 F3:FF:B0+bp:d8 An=A2 F3:FF:B8+bp:d8 An=A3
TBZ	TBZ (abs16)bp,label	mem8(abs16)&(1<<bp)...PSW If ZF=1, PC+5+d8(label)→PC If ZF=0, PC+5→PC	—	—	—	—	—	0	0	0	●	5	5/4	F5:C0+bp:abs16-l:abs16-h:label
	TBZ (abs24)bp,label	mem8(abs24)&(1<<bp)...PSW If ZF=1, PC+7+d8(label)→PC If ZF=0, PC+7→PC	—	—	—	—	—	0	0	0	●	7	7/6	F3:FE:C0+bp:abs24-l:abs24-m: abs24-h:label
	TBZ (d8,An)bp,label	mem8(An+d8)&(1<<bp)...PSW If ZF=1, PC+4(5)+d8(label)→PC If ZF=0, PC+4(5)→PC	—	—	—	—	—	0	0	0	●	4 4 5 5	5/4 5/4 6/5 6/5	F5:80+bp:d8:label An=A0 F5:88+bp:d8:label An=A1 F3:FF:80+bp:d8:label An=A2 F3:FF:88+bp:d8:label An=A3
TBNZ	TBNZ (abs16)bp,label	mem8(abs16)&(1<<bp)...PSW If ZF=1, PC+5→PC If ZF=0, PC+5+d8(label)→PC	—	—	—	—	—	0	0	0	●	5	5/4	F5:C8+bp:abs16-l:abs16-h:label
	TBNZ (abs24)bp,label	mem8(abs24)&(1<<bp)...PSW If ZF=1, PC+7→PC If ZF=0, PC+7+d8(label)→PC	—	—	—	—	—	0	0	0	●	7	7/6	F3:FE:C8+bp:abs24-l:abs24-m: abs24-h:label
	TBNZ (d8,An)bp,label	mem8(An+d8)&(1<<bp)...PSW If ZF=1, PC+4(5)→PC If ZF=0, PC+4(5)+d8(label)→PC	—	—	—	—	—	0	0	0	●	4 4 5 5	5/4 5/4 6/5 6/5	F5:A0+bp:d8:label An=A0 F5:A8+bp:d8:label An=A1 F3:FF:A0+bp:d8:label An=A2 F3:FF:A8+bp:d8:label An=A3
Bcc	BEQ label	If ZF=1, PC+2+d8(label)→PC If ZF=0, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E8:d8 *18
	BNE label	If ZF=0, PC+2+d8(label)→PC If ZF=1, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E9:d8 *19
	BLT label	If (VF^NF)=1, PC+2+d8(label)→PC If (VF^NF)=0, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E0:d8 *20

Notes: 17* Performed under the conditions of bus lock and disabled interrupts.
18* src=dest (lower 16 bits)
19* src≠dest (lower 16 bits)
20* src>dest (lower 16 bits, signed)

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
Bcc	BLE label	If ((VF^NF) I ZF)=1, PC+2+d8(label)→PC If ((VF^NF) I ZF)=0, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E3:d8 *21
	BGE label	If (VF^NF)=0, PC+2+d8(label)→PC If (VF^NF)=1, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E2:d8 *22
	BGT label	If ((VF^NF) I ZF)=0, PC+2+d8(label)→PC If ((VF^NF) I ZF)=1, PC+2→P	—	—	—	—	—	—	—	—	—	2	2/1	E1:d8 *23
	BCS label	If CF=1, PC+2+d8(label)→PC If CF=0, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E4:d8 *24
	BLS label	If (CF I ZF)=1, PC+2+d8(label)→PC If (CF I ZF)=0, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E7:d8 *25
	BCC label	If CF=0, PC+2+d8(label)→PC If CF=1, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E6:d8 *26
	BHI label	If (CF I ZF)=0, PC+2+d8(label)→PC If (CF I ZF)=1, PC+2→PC	—	—	—	—	—	—	—	—	—	2	2/1	E5:d8 *27
	BVC label	If VF=0, PC+3+d8(label)→PC If VF=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:FC:d8 *28
	BVS label	If VF=1, PC+3+d8(label)→PC If VF=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:FD:d8 *29
	BNC label	If NF=0, PC+3+d8(label)→PC If NF=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:FE:d8 *30
	BNS label	If NF=1, PC+3+d8(label)→PC If NF=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:FF:d8 *31
	BRA label	PC+2+d8(label)→PC	—	—	—	—	—	—	—	—	—	2	2	EA:d8
Bccx	BEQX label	If ZX=1, PC+3+d8(label)→PC If ZX=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E8:d8 *32
	BNEX label	If ZX=0, PC+3+d8(label)→PC If ZX=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E9:d8 *33

Notes: 21* src≥dest (lower 16 bits, signed)
22* src≤dest (lower 16 bits, signed)
23* src<dest (lower 16 bits, signed)
24* src>dest (lower 16 bits, unsigned)
25* src≥dest (lower 16 bits, unsigned)
26* src≤dest (lower 16 bits, unsigned)
27* src<dest (lower 16 bits, unsigned)
28* VF=0
29* VF=1
30* NF=0
31* NF=1
32* src=dest (24 bits)
33* src≠dest (24 bits)

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
Bccx	BLTX label	If (VX^NX)=1, PC+3+d8(label)→PC If (VX^NX)=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E0:d8 *34
	BLEX label	If ((VX^NX) l ZX)=1, PC+3+d8(label)→PC If ((VX^NX) l ZX)=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E3:d8 *35
	BGEX label	If (VX^NX)=0, PC+3+d8(label)→PC If (VX^NX)=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E2:d8 *36
	BGTX label	If ((VX^NX) l ZX)=0, PC+3+d8(label)→PC If ((VX^NX) l ZX)=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E1:d8 *37
	BCSX label	If CX=1, PC+3+d8(label)→PC If CX=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E4:d8 *38
	BLSX label	If (CX l ZX)=1, PC+3+d8(label)→PC If (CX l ZX)=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E7:d8 *39
	BCCX label	If CX=0, PC+3+d8(label)→PC If CX=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E6:d8 *40
	BHIX label	If (CX l ZX)=0, PC+3+d8(label)→PC If (CX l ZX)=1 PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:E5:d8 *41
	BVCX label	If VX=0, PC+3+d8(label)→PC If VX=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:EC:d8 *42
	BVSX label	If VX=1, PC+3+d8(label)→PC If VX=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:ED:d8 *43
	BNCX label	If NX=0, PC+3+d8(label)→PC If NX=1, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:EE:d8 *44
	BNSX label	If NX=1, PC+3+d8(label)→PC If NX=0, PC+3→PC	—	—	—	—	—	—	—	—	—	3	3/2	F5:EF:d8 *45
JMP	JMP label16	PC+3+d16(label16)→PC	—	—	—	—	—	—	—	—	—	3	2	FC:d16-l:d16-h
	JMP label24	PC+5+d24(label24)→PC	—	—	—	—	—	—	—	—	—	5	4	F4:E0:d24-l:d24-m:d24-h
	JMP (An)	An→PC	—	—	—	—	—	—	—	—	—	2	3	F0:An<<2

Notes: 34* src>dest (24 bits, signed)
35* src≥dest (24 bits, signed)
36* src≤dest (24 bits, signed)
37* src<dest (24 bits, signed)
38* src>dest (24 bits, unsigned)
39* src≥dest (24 bits, unsigned)
40* src≤dest (24 bits, unsigned)
41* src<dest (24 bits, unsigned)
42* VX=0
43* VX=1
44* NX=0
45* NX=1

Instruction	Mnemonic	Operation	OP EX.	Flag								Code Size	Cycle	Machine Code
				VX	CX	NX	ZX	VF	CF	NF	ZF			
JSR	JSR label16	A3-4→A3 PC+3→mem24(A3) PC+3+d16(label16)→PC	—	—	—	—	—	—	—	—	—	3	4	FD:d16-l:d16-h
	JSR label24	A3-4→A3 PC+5→mem24(A3) PC+5+d24(label24)→PC	—	—	—	—	—	—	—	—	—	5	5	F4:E1:d24-l:d24-m:d24-h
	JSR (An)	A3-4→A3 PC+2→mem24(A3) An→PC	—	—	—	—	—	—	—	—	—	2	5	F0:01+An<<2
NOP	NOP	PC+1→PC	—	—	—	—	—	—	—	—	—	1	1	F6
RTS	RTS	mem24(A3)→PC A3+4→A3	—	—	—	—	—	—	—	—	—	1	5	FE
RTI	RTI	mem16(A3)→PSW mem24(A3+2)→PC A3+6→A3	—	●	●	●	●	●	●	●	●	1	6	EB
PXST	PXST	Prefix instruction reversing the following instruction of addition/subtraction on saturation operation flag of PSW	—	—	—	—	—	—	—	—	—	2	2	F3:FC

Reading the instruction set

■ Symbols used in tables

Dn, Dm, Di	Data register
An, Am	Address register
MDR, PSW, PC	Multiply/Divide Register, Processor Status Word, Program Counter
imm8, imm16, imm16-l, imm16-h	Constant
imm24, imm24-l, imm24-m, imm24-h	
d8, d16, d16-l, d16-h	Displacement
d24, d24-l, d24-m, d24-h	
abs16, abs16-l, abs16-h	Absolute address
abs24, abs24-l, abs24-m, abs24-h	
mem8 (An), mem8 (abs16), mem8 (abs24)	8-bit memory data which is determined by the address inside parentheses ( )
mem16 (An), mem16 (abs16), mem16 (abs24)	16-bit memory data which is determined by the address inside parentheses ( )
mem24 (Am), mem24 (abs16), mem24 (abs24)	24-bit memory data which is determined by the address inside parentheses ( )
.bp,.lsb, .msb	Bit specification
&, I , ^	Logical AND, logical OR, exclusive OR
~, <<, >>	Bit inversion, bit shift
VX, CX, NX, ZX,	Extended overflow flag, carry flag, negative flag, zero flag (24-bit data)
VF, CF, NF, ZF	Overflow flag, carry flag, negative flag, zero flag (16-bit data)
temp	CPU internal temporary register
→ , ...	Substitution, reflects calculation results

■ OP EX. (Operand Extensions)

0	Zero-extension
S	Sign-extension
—	Not applicable

■ Flag

●	Changes
—	No change
0	Always 0
1	Always 1
?	Undefined

■ Code Size

Units : bytes

■ Cycle

Minimum cycle count is shown.
(when quick decoder disabled)
Units : machine cycles

a/b : a cycles if branch taken
b cycles if branch not taken

■ Machine Code

":" indicates a delimiter between bytes.<<2 indicates a 2-bit shift.

Dn, Dm, Di, An, Am : Register numbers

D0	00	A0	00
D1	01	A1	01
D2	10	A2	10
D3	11	A3	11

■ Notes

- 16-bit or 24-bit access instruction must not access odd memory addresses.
- 8-bit displacements (d8) and 16-bit displacements (d16) are all sign-extended.

MN102H SERIES INSTRUCTION MAP

First byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	MOV Dm, (An)															
1	MOVB Dm, (An)															
2	MOV (An), Dm															
3	MOVBU (An), Dm															
4	MOV Dm, (d8, An)															
5	MOV Am, (d8, An)															
6	MOV (d8, An), Dm															
7	MOV (d8, An), Am															
8	MOV Dn, Dm, (when src=dest,MOV imm8, Dn)															
9	ADD Dn, Dm															
A	SUB Dn, Dm															
B	EXTX Dn				EXTXU Dn				EXTXB Dn				EXTXBU Dn			
C	MOV Dn, (abs16)				MOVB Dn, (abs16)				MOV (abs16),Dn				MOVBU (abs16),Dn			
D	ADD imm8, An				ADD imm8, Dn				CMP imm8, Dn				MOV imm16, An			
E	BLT label	BGT label	BGE label	BLE label	BCS label	BHI label	BCC label	BLS label	BEQ label	BNE label	BRA label	RTI	CMP imm16, An			
F	Extended code A	Extended code B	Extended code C	Extended code D	Extended code E	Extended code F	NOP	Extended code G	MOV imm16, Dn				JMP label16	JSR label16	RTS	

Extended Code A

Second byte (Byte 1: F0)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	JMP (A0)	JSR (A0)			JMP (A1)	JSR (A1)			JMP (A2)	JSR (A2)			JMP (A3)	JSR (A3)		
1																
2	BSET Dm, (An)															
3	BCLR Dm, (An)															
4	MOVB (Di, An), Dm															
5																
6																
7																
8	MOVBU (Di, An), Dm															
9																
A																
B																
C	MOVB Dm, (Di, An)															
D																
E																
F																

Extended Code B

Second byte (Byte 1: F1)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4		MOV (Di, An), Dm														
5																
6																
7																
8																
9																
A																
B																
C		MOV Dm, (Di, An)														
D																
E																
F																

Extended Code C

Second byte (Byte 1: F2)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		ADD Dm, An														
1		SUB Dm, An														
2		CMP Dm, An														
3		MOV Dm, An														
4		ADD An, Am														
5		SUB An, Am														
6		CMP An, Am														
7		MOV An, Am														
8		ADDC Dn, Dm														
9		SUBC Dn, Dm														
A																
B																
C		ADD An, Dm														
D		SUB An, Dm														
E		CMP An, Dm														
F		MOV An, Dm														

Extended Code D

Second byte(Byte 1: F3)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	AND Dn, Dm															
1	OR Dn, Dm															
2	XOR Dn, Dm															
3	ROL Dn				ROR Dn				ASR Dn				LSR Dn			
4	MUL Dn, Dm															
5	MULU Dn, Dm															
6	DIVU Dn, Dm															
7																
8																
9	CMP Dn, Dm															
A																
B																
C	MOV D0, MDR	EXT D0			MOV D1, MDR	EXT D1			MOV D2, MDR	EXT D2			MOV D3, MDR	EXT D3		
D	MOV D0, PSW 1*				MOV D1, PSW 1*				MOV D2, PSW 1*				MOV D3, PSW 1*			
E	MOV MDR, Dn				NOT Dn											
F	MOV PSW, Dn												PXST		Extended code H	Extended code I

Notes:1* In case of using this instruction, a value of PSW or the one that has been pushed in stack should not be referred in the interrupt processing program.

Extended Code E

Second byte (Byte 1: F4)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	MOV Dm, (d24, An)															
1	MOV Am, (d24, An)															
2	MOVB Dm, (d24, An)															
3	MOVX Dm, (d24, An)															
4	MOV Dn, (abs24)				MOVB Dn, (abs24)							BSET imm8,(abs24)				BCLR imm8,(abs24)
5	MOV An, (abs24)															
6	ADD imm24, Dn				ADD imm24, An				SUB imm24, Dn				SUB imm24, An			
7	MOV imm24, Dn				MOV imm24, An				CMP imm24, Dn				CMP imm24, An			
8	MOV (d24, An), Dm															
9	MOVBU (d24, An), Dm															
A	MOVB (d24, An), Dm															
B	MOVX (d24, An), Dm															
C	MOV (abs24), Dn				MOVB (abs24), Dn				MOVBU (abs24), Dn							
D	MOV (abs24), An															
E	JMP label24	JSR label24		BSET imm8,(abs16)				BCLR imm8,(abs16)	BSET imm8, (d8,An)				BCLR imm8, (d8,An)			
F	MOV (d24, An), Am															

Extended Code F

Second byte(Byte 1: F5)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	AND imm8, Dn				BTST imm8, Dn				OR imm8, Dn				ADDNF imm8, An			
1	MOVB Dm, (d8, An)															
2	MOVB (d8, An), Dm															
3	MOVBU (d8, An), Dm															
4	Extended Code J															
5	MOVX Dm, (d8, An)															
6	Extended Code K															
7	MOVX (d8, An), Dm															
8	TBZ(d8, A0) bp,label								TBZ(d8, A1) bp,label							
9	BSET(d8, A0) bp								BSET(d8, A1) bp							
A	TBNZ(d8, A0) bp,label								TBNZ(d8, A1) bp,label							
B	BCLR(d8, A0) bp								BCLR(d8, A1) bp							
C	TBZ(abs16) bp,label								TBNZ(abs16) bp,label							
D	BSET(abs16) bp								BCLR(abs16) bp							
E	BLTX label	BGTX label	BGEX label	BLEX label	BCSX label	BHIX label	BCCX label	BLSX label	BEQX label	BNEX label			BVCX label	BVSX label	BNCX label	BNSX label
F	Extended Code L												BVC label	BVS label	BNC label	BNS label

Extended Code G

Second byte (Byte 1: F7)

Second byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	AND imm16, Dn				BTST imm16, Dn				ADD imm16, An				SUB imm16, An			
1	AND imm16 PSW				OR imm16 PSW				ADD imm16, Dn				SUB imm16, Dn			
2	MOV An, (abs16)															
3	MOV (abs16), An															
4	OR imm16, Dn								CMP imm16, Dn				XOR imm16, Dn			
5	MOVBU (d16, An), Dm															
6	MOVX Dm ,(d16, An)															
7	MOVX (d16, An), Dm															
8	MOV Dm, (d16, An)															
9	MOVB Dm, (d16, An)															
A	MOV Am, (d16, An)															
B	MOV (d16, An), Am															
C	MOV (d16, An), Dm															
D	MOVB (d16, An), Dm															
E																
F																

Extended Code H

Third byte (Byte 1: F3, Byte 2 : FE)

Third byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C	TBZ (abs24)bp,label								TBNZ (abs24)bp,label							
D	BSET (abs24)bp								BCLR (abs24)bp							
E																
F																

Extended Code I

Third byte (Byte 1: F3, Byte 2 : FF)

Third byte Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8	TBZ (d8,A2)bp,label								TBZ(d8,A3)bp,label							
9	BSET(d8,A2)bp								BSET (d8,A3)bp							
A	TBNZ (d8,A2)bp,label								TBNZ(d8,A3)bp,label							
B	BCLR(d8,A2)bp								BCLR (d8,A3)bp							
C																
D																
E																
F																

Extended Code J
Third byte (Byte 1: F5, Byte 2 : 4n)

Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	MULQL Dn,Dm	MULQH Dn,Dm														
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Extended Code K
Third byte (Byte 1: F5, Byte 2 : 6n)

Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1	MULQ Dn,Dm															
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Extended Code L
Third byte (Byte 1: F5, Byte 2 : Fn)

Upper/Lower	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					MULQL imm8,Dn	MULQH imm8,Dn			MULQL imm16,Dn	MULQH imm16,Dn						
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Ver2.1 (2001.03.15)

Record of Changes

MN102H74G/74F/74D/F74G LSI User's Manual Record of Changes (Ver.1.0 to Ver.1.4)

Page	Section	Definition	Details of Changes	
			PreviousEdition (Ver.1.0)	New Edition (Ver.1.4)
All chapters	-	Change	MN102H74G/74F/F74G	MN102H74G/74F/74D/F74G
I-7	Figure 1-1-2	Addition	-	MN102H74D: 64 KB
	Table 1-1-1	Addition	ROM Capacity Single-chip Mode : 96KB/128KB Memory Expansion Mode :	ROM Capacity Single-chip Mode : 64KB/96KB/128KB Memory Expansion Mode : 128 KB
III-3	Figure 3-1-2	Change	Internal ROM : 128 KB	Internal ROM : MAX 128 KB
VI-2	Table 6-1-2	Addition	-	Maximum Baud Rate/Clock Synchronous Mode : 3 Mbps Note) For sequence synchronous reception in serial interface 2, 3 using external clock (SBTn : n =2,3) at transmission rate over 1.5 Mbps, keep an interval at least 4 machine cycles (at app.333 ns, 12 MHz oscillator) between data receptions.
VIII-11	Caution	Addition	-	Single address mode is enable only in transfer between USB-FIF0 and the internal RAM.
	Figure 8-5-6	Change	NWEL is input as NRE in the USB core, memory output is transferred directly without the CPU.	It transfers USB core output directly without using the CPU, as NWEL is input as NRE to USB core.
IX-2	Table 9-1-1	Change	Address Mode Note 1: Single address mode (only between USB-FIFO and memory)	Address Mode Note 1: Single address mode (only between USB-FIFO and internal RAM)
IX-6	ATnA MD[1:0]	Change	When the single address mode is selected, the source address or the destination address is determined based on the memory.	When the single address mode is selected, the source address or the destination address should be set based on the addressing in the internal RAM.
IX-9	Caution	Addition	-	Select early write (ATnWMD[1:0] = 10) for write access to the internal RAM in single address mode. Other write operation is disable.
IX-20	Line 3-4	Change	The single address mode is used for the high-speed transfer between USB-FIFO and memory.	The single address mode is used for the high-speed transfer between USB-FIFO and the internal RAM.
IX-21	Line 1-4	Change	In the single address mode, the data is transferred between memory and USB-FIFO which has the special interface corresponded to the single address mode in 1 bus transaction. When the memory is used as the source, the single address mode with source address specification is selected. On the other hand, when the memory is used as the destination, the single address mode with destination address specification is selected.	In the single address mode, the data is transferred between the internal RAM and USB-FIFO which has the special interface corresponded to the single address mode in 1 bus transaction. When the internal RAM is used as the source, the single address mode with source address specification is selected. On the other hand, when the internal RAM is used as the destination, the single address mode with destination address specification is selected.
IX-33	Line 4	Addition	-	In single address mode, either source address or destination address is specified depending on which side is the internal RAM set. Also the bus mode operation is determined depending on the specified address side.
IX-33	Line 5, Ex) 5-1	Change	In example 5-1 the data is transferred from the 16-bit memory to 8-bit device (USB-FIFO) in single address mode.	In example 5-1 the data is transferred from the internal RAM to 8-bit device (USB-FIFO) in single address mode.
IX-33	Line 6, Ex) 5-2	Change	In example 5-2, the data is transferred from 8-bit device (USB-FIFO) to the 16-bit memory in single address mode.	In example 5-2, the data is transferred from 8-bit device (USB-FIFO) to the internal RAM in single address mode.
XII-4	Table	Addition	-	C. Electrical Characteristics (1) DC Characteristics MN102H74D : α= 0

MN102H74G/74F/74D/F74G
LSI User's Manual

February, 2002 1st Edition 4th Printing

Issued by Matsushita Electric Industrial Co., Ltd.

© Matsushita Electric Industrial Co., Ltd.

Semiconductor Company, Matsushita Electric Industrial Co., Ltd.

Nagaokakyo, Kyoto 617-8520, Japan
Tel: (075) 951-8151
http://www.panasonic.co.jp/semicon/

SALES OFFICES

■ NORTH AMERICA

●U.S.A. Sales Office:

Panasonic Industrial Company [PIC]

• New Jersey Office:
Two Panasonic Way Secaucus, New Jersey 07094 U.S.A.
Tel: 1-201-348-5257 Fax:1-201-392-4652

• Chicago Office:
1707 N. Randall Road Elgin, Illinois 60123-7847 U.S.A.
Tel: 1-847-468-5720 Fax:1-847-468-5725

• Milpitas Office:
1600 McCandless Drive Milpitas, California 95035 U.S.A.
Tel: 1-408-942-2912 Fax:1-408-946-9063

• Atlanta Office:
1225 Northbrook Parkway Suite 1-151 Suwanee, GA 30024 U.S.A.
Tel: 1-770-338-6953 Fax:1-770-338-6849

• San Diego Office:
9444 Balboa Avenue, Suite 185, San Diego, California 92123 U.S.A.
Tel: 1-619-503-2903 Fax:1-858-715-5545

●Canada Sales Office:

Panasonic Canada Inc. [PCI]
5770 Ambler Drive 27 Mississauga, Ontario, L4W 2T3 CANADA
Tel: 1-905-238-2101 Fax:1-905-238-2414

■ LATIN AMERICA

●Mexico Sales Office:

Panasonic de Mexico, S.A. de C.V. [PANAMEX]
Amores 1120 Col. Del Valle Delegacion Benito Juarez C.P. 03100 Mexico, D.F. MEXICO
Tel: 52-5-488-1000 Fax:52-5-488-1073

• Guadalajara Office:
SUCURSAL GUADALAJARA
Av. Lazaro Cardenas 2305 Local G-102 Plaza Comercial Abastos; Col. Las Torres Guadalajara, Jal. 44920 MEXICO
Tel: 52-3-671-1205 Fax:52-3-671-1256

●Brazil Sales Office:

Panasonic do Brasil Ltda. [PANABRAS]
Caixa Postal 1641, Sao Jose dos Campos, Estado de Sao Paulo
Tel: 55-12-335-9000 Fax:55-12-331-3789

■ EUROPE

●U.K. Sales Office:

Panasonic Industrial Europe Ltd. [PIEL]
Willoughby Road, Bracknell, Berks., RG12 8FP, THE UNITED KINGDOM
Tel: 44-1344-85-3671 Fax:44-1344-85-3853

●Germany Sales Office:

Panasonic Industrial Europe GmbH [PIEG]
Hans-Pinsel-Strasse 2 85540 Haar, GERMANY
Tel: 49-89-46159-119 Fax:49-89-46159-195

■ ASIA

●Singapore Sales Office:

Panasonic Semiconductor of South Asia [PSSA]
300 Beach Road, #16-01, The Concourse, Singapore 199555 THE REPUBLIC OF SINGAPORE
Tel: 65-390-3688 Fax:65-390-3689

●Malaysia Sales Office:

Panasonic Industrial Company (M) Sdn. Bhd. [PICM]

• Head Office:
Tingkat 16B, Menara PKNS Petaling Jaya, No.17, Jalan Yong Shook Lin 46050 Petaling Jaya, Selangor Darul Ehsan, MALAYSIA
Tel: 60-3-7951-6601 Fax:60-3-7954-5968

• Penang Office:
Suite 20-07,20th Floor, MWE Plaza, No.8, Lebuh Farquhar,10200 Penang, MALAYSIA
Tel: 60-4-201-5113 Fax:60-4-261-9989

• Johore Sales Office:
Menara Pelangi, Suite8.3A, Level8, No.2, Jalan Kuning Taman Pelangi, 80400 Johor Bahru, Johor, MALAYSIA
Tel: 60-7-331-3822 Fax:60-7-355-3996

●Thailand Sales Office:

Panasonic Industrial (THAILAND) Ltd. [PICT]
252-133 Muang Thai-Phatra Complex Building, 31st Fl. Rachadaphisek Rd., Huaykwang, Bangkok 10320, THAILAND
Tel: 66-2-693-3428 Fax:66-2-693-3422

●Philippines Sales Office: [PISP]

Panasonic Indsutrial Sales Philippines Division of Matsushita Electric Philippines Corporation
102 Laguna Boulevard,Bo.Don Jose Laguna Technopark, Santa. Rosa, Laguna 4026 PHILIPPINES
Tel: 63-2-520-8615 Fax:63-2-520-8629

●India Sales Office:

National Panasonic India Ltd. [NPI]
E Block, 510, International Trade Tower Nehru Place, New Delhi_110019 INDIA
Tel: 91-11-629-2870 Fax:91-11-629-2877

●Indonesia Sales Office:

P.T.MET & Gobel [M&G]
JL. Dewi Sartika (Cawang 2) Jakarta 13630, INDONESIA
Tel: 62-21-801-5666 Fax:62-21-801-5675

●China Sales Office:

Panasonic Industrial (Shanghai) Co., Ltd. [PI(SH)]
Floor 6, Zhong Bao Mansion, 166 East Road Lujian Zui, PU Dong New District, Shanghai, 200120 CHINA
Tel: 86-21-5866-6114 Fax:86-21-5866-8000

Panasonic Industrial (Tianjin) Co., Ltd. [PI(TJ)]
Room No.1001, Tianjin International Building 75, Nanjin Road, Tianjin 300050, CHINA
Tel: 86-22-2313-9771 Fax:86-22-2313-9770

Panasonic SH Industrial Sales (Shenzhen) Co., Ltd. [PSI(SZ)]
7A-107, International Bussiness & Exhibition Centre, Futian Free Trade Zone, Shenzhen 518048, CHINA
Tel: 86-755-359-8500 Fax:86-755-359-8516

Panasonic Shun Hing Industrial Sales (Hong Kong) Co., Ltd. [PSI(HK)]
11th Floor, Great Eagle Center 23 Harbour Road, Wanchai, HONG KONG
Tel: 852-2529-7322 Fax:852-2865-3697

●Taiwan Sales Office:

Panasonic Industrial Sales (Taiwan) Co.,Ltd. [PIST]

• Head Office:
6F, 550, Sec. 4, Chung Hsiao E. RD. Taipei, 110, TAIWAN
Tel: 886-2-2757-1900 Fax:886-2-2757-1906

• Kaohsiung Office:
6th Floor, Hsin Kong Bldg. No.251, Chi Hsien 1st Road Kaohsiung 800, TAIWAN
Tel: 886-7-346-3815 Fax:886-7-236-8362

●Korea Sales Office:

Panasonic Industrial Korea Co., Ltd. [PIKL]
Kukje Center Bldg. 11th Fl., 191 Hangangro 2ga, Youngsan-ku, Seoul 140-702, KOREA
Tel: 82-2-795-9600 Fax:82-2-795-1542

230102
Printed in JAPAN

© Matsushita Electric Industrial Co., Ltd. 2002